INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Mt-Djing: Multitouch DJing Table

Pedro Lopes

Dissertation for the degree of Master in
Information Systems and Computer Engineering

Júri

Chairman:	Professor Doutor José Manuel Nunes Salvador Tribolet
Adviser:	Professor Doutor João António Madeiras Pereira
Co-adviser:	Professor Doutor Alfredo Manuel dos Santos Ferreira Junior
Jury Member:	Professor Doutor Nuno Manuel Robalo Correia

October 2010

Acknowledgements

As DJ history itself, this thesis has been a long journey, full of turning points and milestones. The astonishing number of people I would like to acknowledge shows how numerous were the dark alleys and obstacles I had to transpose during this last year.

Firstly, to my father and mother for outstanding support, not only through this work but for the two decades before.

To my advisor and co-advisor, Prof. João Madeiras Pereira and Prof. Alfredo Ferreira Jr., for strong belief on Mt-Djing from square one. Moreover I kindly acknowledge the contribution of my colleague Diogo Mariano, whom I thank for all the support given, endless brainstorming sessions and late work nights.

A word of appreciation to Ricardo Jota and Bruno de Araújo, for the interest and support to this work, shown in the form of the many conversations exchanged upon this subject. Other colleagues have their place within this thank-you note, namely Tiago Ribeiro and Guilherme Fernandes, for their help in reviewing and discussing the contents.

And at last, but (definitely) not least, to all DJs and accompanying group, that took the time and effort to contribute so much to this work - that ultimately is theirs more than it is mine - specially: Pedro Farinha, Pedro Sousa (aka Alfredo Carajillo), João Gomes (aka Mush Von Namek), Igor Sousa and Nuno Moita.

This work was funded in part by the European Commission's Seventh Framework Programme (FP7/2007-2013) through project MAXIMUS, grant agreement IST-2007-1-217039 and through FCT MIVIS PTDC/EIA/104031/2008.

Lisboa, October 2010

Pedro Lopes

To the role of DJs in modern music.
Towards becoming part of music
itself.

Resumo

A história do DJing encontra-se repleta de inovações ao nível tecnológico. Contudo, apesar destes dispositivos permitirem novas funcionalidades aos DJs, existe uma lacuna em produtos que ofereçam uma interacção táctil sob uma aplicação de DJing Virtual, encontrando-se estas tipicamente reduzidas aos dispositivos de entrada tradicionais.

Recentemente diversas propostas endereçam tópicos co-relacionados com a actividade do DJing; contudo, as metáforas apresentadas não são coerentes com o léxico gestual do DJ, exigindo um extenso período de treino.

Através do envolvimento de um grupo de DJs peritos, desenvolvemos uma solução multi-toque que possibilita o mapeamento directo dos conceitos fundamentais do modelo mental dos DJs numa aplicação Virtual. Na solução proposta, a interacção gestual do *setup* Tradicional apresenta-se coerente com as metáforas desenhadas; e herdam-se as vantagens da virtualização do domínio do DJing. O sistema aborda os principais requisitos do DJ contemporâneo, como identificado na nossa investigação, salientando-se: robustez, baixa-latência, controlo externo, adaptabilidade, suporte a *plugins* áudio e assente em normas para a comunicação com soluções existentes.

A solução proposta permite ainda melhorias no desempenho das tarefas de DJing, contribuindo com uma interface totalmente reconfigurável em tempo real e adaptável às necessidades do utilizador. O sistema permite aos DJs exercerem a sua criatividade, criando cenários de uma forma natural, que de outra maneira não seriam possíveis nos *setups* típicos. A comparação da nossa proposta relativamente aos *setups* Virtuais, Tradicionais e Híbridas comprova a adequabilidade de uma superfície multi-toque para utilizadores experientes que desejem abandonar o *setup* Virtual em prol duma interacção mais natural.

Abstract

Disc-jockeys have come a long way, through technological evolutions. This path led them to the status and recognition they have achieved in our society. But as impressive as those technological evolutions are, as far as DJing is concerned, there are still few applications that support hands-on interaction over virtual DJ applications, and those are typically reduced to the traditional input devices. Furthermore, recent proposals apply new DJ metaphors, but are not successful in maintaining coherence with traditional DJ lexicon of gestures and concepts, thus requiring an extensive learning period.

Accounting user feedback from an accompanying group of DJ experts, we devised a multitouch solution that maps physical core DJ concepts into a virtual application. In the proposed solution, gestural interaction from Traditional Setups remains coherent while inheriting advantages from the virtualization of the DJing domain. The system addresses typical requirements of contemporary DJing, as identified by our research, namely: robustness, low-latency, external control, adaptability, audio plugins and connectivity standard-compliance.

Additionally, we support task improvements, such as dynamic re-routing of music flow, seamlessly merging edit with DJ-performance mode and rearrangement of interface layout according to user needs. Our system allows DJs to exercise creativity with a natural interaction, creating scenarios that are not possible in the real world. Comparison against Traditional, Virtual and Hybrid DJ setups showed how a multitouch DJing surface developed with DJ involvement, can suit experienced users changing to Virtual setups.

Palavras Chave
Keywords

Palavras Chave

Interacção Multi-toque

DJing

Concepção Centrada no Utilizador

Interface Natural de Utilizador

Keywords

Multitouch Interaction

DJing

User-centered Design

Natural User Interface

Index

List of Figures

List of Tables

Acronyms

ACM Association for Computing Machinery

AD Analogue to Digital

API Application Programming Interface

BPM Beats per Minute

CPU Central Processing Unit

DA Digital to Analogue

DJ Disc Jockey

FTIR Frustrated Total Internal Reflection

GUI Graphical User Interface

IRCAM Institut de Recherche et Coordination Acoustique/Musique

MIDI Musical Instrument Digital Interface

NUI Natural User Interface

OSC Open Sound Control

RPM Rotations per Minute

UI User Interface

VU Volume Units

WYSIWYG What you see is what you get

1 Introduction

1.1 Context and Motivation

The contemporary DJ job expands far beyond selecting records at a party and its journey has always been teamed up with technological advancements: the turntable, the DJ mixer, the CD and MP3 players and the new state-of-the-art DJing systems - a continuum of evolution both in technology and musical impact [1; 2].

These DJ tools, commonly called DJ Setups [3], play an explicit role in defining the DJ style, and are directly related to user's needs. Throughout our research we used a classification based upon their interaction and technological idiosyncrasies and identified three major setups: Traditional, Virtual and Hybrid.

Chronologically speaking, the three DJ setups considered here represent different evolution stages of DJing tools. The Traditional setup, introduced more than three decades ago, utilises analogue devices, typically two turntables and a signal mixer [3]. This gear allows users to exercise direct bimanual interaction on the hardware, but forces them to travel with complex and heavy equipment that, in addition, requires technical maintenance.

Later on, the Traditional setup was virtualized into a software application, hence denoted Virtual setup. DJing applications in a Virtual setup have a handful of features that are not available in the Traditional, that we identified through interviews with our accompanying group of DJs: providing digital audio processing, audio plug-in integration, unlimited tracks, and weightless environment, to name a few. However, they are also heavily criticized not only for their non-natural interaction, based on traditional input devices such as mouse, keyboard or dedicated hardware controllers, but also their high learning curves, specially to users acquainted with traditional DJ gestures.

Recently, efforts have been carried out in order to minimize drawbacks in Virtual setups, culminating in the Hybrid, an union of the Traditional and Virtual setups. The Hybrid provides a digital control system, allowing the user to operate a DJ application by direct manipulation over traditional components. But, although it solves the non-natural mapping problem found in earlier Virtual setups, it also triggered, once again, the need for analogue equipment, along with its limited features and high acquisition, maintenance, and transportation costs.

1.2 Problem Statement

We identified that, not only Traditional and Hybrid DJ solutions have hardware requirements that are inadequate to support the new tools that modern DJs praise, they also have high costs in acquisition, maintenance and transportation, driving various users to switch over to DJing software products. Virtual setups, on the other hand, include exciting features in terms of musical expression and extensiveness, but are bounded to a non-natural interaction scheme, derived from exercising indirect control via input devices.

1.3 Contributions of our work

To overcome the problems identified above, we propose a multitouch interactive DJing application (where user-feedback from an accompanying group of DJ experts is accounted for), one that bridges the gap between Traditional and Virtual setups.

The proposed system merges the benefits of Virtual DJing applications with natural interaction found in Traditional DJing setups, rather than relying on typical input devices. Additionally, digital audio manipulation enables us not only to improve the DJ's performance but also allows him/her to exercise creativity, creating custom setups that are not possible in traditional live situations. As presented throughout this document, there are related works with a similar scope, but none actually addresses the touch-design challenges in DJ tasks to ensure hands-free interaction.

Our proposal targets DJs in need of a direct interface to exercise bimanual real-time control over their performances. Furthermore, due to its modular architecture, it may be used both as a typical DJ setup (playing songs, mixing, and so forth) and as an external controller for the upcoming, modern DJ artist, allowing our solution to be easily extended to any existing DJ setup.

Finally, a novel evaluation was conducted by a panel of DJ experts, in which our prototype was compared against all three standard DJ setups. Ultimately this concludes upon the adequacy of multitouch in the DJ context, producing a set of hands-free interaction metaphors closely related to the DJs' mental model.

1.4 Publications

Throughout the timespan of this work three papers have been published. The first two, **Interacção 2010** and **ACE 2010**, respectively present challenges in multitouch DJing and report an evaluation of our proposal with a group of DJ experts. The last one, published at **RecPad 2010** by the same author,

proposes a new footcontroller to enhance bimanual interaction on our DJ application. The references of these contributions are as follows:

- Lopes, P., Ferreira, A., Pereira, J.: Interacção multi-toque no contexto do DJing. In: Interacção 2010: 4ª Conferência Interacção Pessoa-Máquina (2010).
- Lopes, P., Ferreira, A., Pereira, J.: Multitouch DJing surface. In: ACE '10: Proceedings of the 2010 ACM SIGCHI international conference on Advances in Computer Entertainment Technology, ACM (2010).
- Lopes, P., Fernandes, G., Jorge, J.A.: Trainable DTW-based classifier for recognizing feet-gestures. In: RECPAD 2010: Proceedings of the 16th Portuguese Conference on Pattern Recognition. (2010)

Additionally, two more submissions are already undergoing: one will include the related work section, a survey of new trends in DJ tools, and will be submitted to the **Journal of New Music Research**; another will report the DJ's gestural evaluation presented in section 4.3.4 and will be submitted to a conference of the HCI thematic area, still undefined due to the project's calendar.

1.5 Outline

This dissertation is structured into seven chapters. The first (and current) chapter presents an introduction, identifying the problem and highlighting our solution and contributions. In chapter two we provide the background to understand the core concepts of the DJ's mental model, while in chapter three we discuss related work that is relevant to our proposal. Throughout the fourth chapter we propose an architecture that fulfils DJ's requirements and elaborate on the interface concepts of our proposal, focusing on designing natural interaction metaphors to DJ tasks. We follow it with an user evaluation, presented in the fifth chapter, where we draw a critical comparison of our proposal against standard solutions, using data collected from two test sessions conducted with DJ experts. Finally, in chapters six and seven, we draw the final conclusions, summarize our work, and present some future steps that could be taken to improve upon these results.

2 Background

So the story can be said to begin with Berliner in 1894, or Edison in 1877, or even John Cage in 1939. But what really makes this history worthwhile is what happened in the Bronx in the mid-seventies.

Kjetil Falkenberg Hansen *in Turntablisme - His Master's Voice: The Art of the Record Player [4]*

In this chapter, we briefly address the history of DJing, with an emphasis on those tools that came along and revolutionised the way DJs work. First, we draw our attention to the setup that paved the way to the all-time standard analogue system: a single mixer with two turntables; then, we move along the timeline, looking at some other landmarks. Finally, based on DJ-specific literature and interviews with the accompanying group of DJs as to what are the major drivers to DJ performance, we draw some conclusions on the requirements of typical DJ setups.

2.1 Historical Developments on Djing

The art of DJing has surely come a long way, starting up in the early days of Radio Waves and in the first shows using records[1], partly due to Reginal Fessenden (and many other inventors), who in 1906 broadcasted a record on the air for the first time. Then, the golden years of ballroom and pop music came along, and DJs' importance rose to new heights, along with their social status. Radio DJs became the most active and important figures of music promotion (a role that DJs still carry today, and will probably live on).

One could argue that turntables were very early regarded as an instrument, through the musical experimentations of *avant-garde* pioneers such as John Cage and Pierre Shaffer circa 40-50's [4; 1]. Both composers were interested in its reproduction and manipulation potential, but as history dictated, turntable usage by *musique concrète* fell, under the domination of magnetic tape [5]. DJing went through a series of mutations, both social and technical, including DJs appearances at TV shows and ballrooms,

[1] Dr. Elman B. Meyers had an 18 hour radio show with almost just records, a classical example of pure-record broadcasting [2].

and advancements on audio reproduction gear and vinyl pressing formats[2], to name a few. However, the "boom" of a new DJ culture, one that still spans onto the 21st century and shaped the DJ of today as more than mere musical collectors, only occurred later on.

Almost at the same time, in two geographically sparse locations, Jamaica and the Bronx, the new era of DJing was witnessing its birth [2]. On the tropics, the selectors [3] were altering the records in studio, pressing new versions (called dubs) which only used modified instrumental parts. Although this is common practice in today's current DJ culture, it then signaled the beginning of an era: pressing music specifically for mixing.

2,500 kilometers to the north, on the declining NYC borough, Hip-Hop was witnessing its birth, largely due to the black movement derived from 60's and 70's music (specially funk, a possible musical precursor of Hip-Hop as a musical expression - "rap" [6; 7]). With leading front men like Afrika Bambataa, Kool Herc, Grand Mixer DXT [4] and other artists, such as Grand Wizard Theodore (who's credited the invention of the scratch), the time was right for turntablism and to the discovery of a new instrument: the turntable [2; 4].

To the non-familiarized reader, some explanation is required: Scratching is a form of musical expression performed on the turntable [8; 9]. The concept is fairly simple, but results can be amazingly rich both harmonically and rhythmically (see Appendix B for a selected discography of DJ music). In plain words, the DJ manipulates the audio in realtime, moving the record back and forth with his hand. Furthermore, he mutes and "opens" the volume with the help of the DJ mixer, something that gives the scratching technique more rhythmic complexity because a longer sound can be cut in several shorter ones.

Many of the DJ techniques that are used today were created within these two DJ movements, namely "the reverse spin", which is performed by audibly reversing the playback of the record with a hand movement, and another, probably the most relevant of all techniques, "beat-matching" - the core skill of almost every dance music DJ, where the goal is to sync two different music songs to create a new composition. Beat-matching two different songs is achieved using very simple smaller techniques, developed from the radio days and the early DJs, such as "mixing" (adding two sound sources simultaneously), "pre-listening" (listening to a record with the purpose of finding a specific point, without the signal going to the audience) and specially "slip-cueing" (invented by DJ Francis Grasso [2]). The slip-cueing technique allows a DJ to start the record at a more precise timing, making it easier to synchronise with the running song. In order to accomplish it, the record is stopped with the hand but, as the turntable

[2] Evolutions in recorded-music reproduction lead the way through the first 78 RPM shellac records, to the 45 and 33 RPM vinyl formats, both still in commercial usage.

[3] "Selector" is a common name for the reggae/dancehall DJ [2].

[4] Grand Mixer DXT performed the very first public display of scratching to a massive worldwide audience, integrating the act of fusion-jazz pianist Herbie Hancock as seen on Figure 2.1, winning one 1984 Grammy Award for the "Future Shock" album.

Figure 2.1: Grand Mixer DXT performing live with Herbie Hancock's band in 1984. [12]

platter underneath continues spinning, the record can be launched almost instantly, just by removing the hand from the record [2; 10; 11].

There's also a more extreme lexicon of techniques [13] developed by the so-called turntablists (sound manipulators that use the turntable as an instrument [14; 4]), such as the already mentioned "scratch" and "beat-juggling", a fast mixing, in two opposite turntables, of two equal musical samples that extends the sample duration by looping it manually [15]. All these "skills" and actions define a vocabulary for the modern DJ and a set of gestures that he/she performs regularly [9; 16]. Although scratch skills are embedded in the scope of turntablists [5] (see Appendix D for a more in-depth terminology and Appendix B for DJ-style classification), understanding this lexicon will help us in our effort to convey a natural interface with adequate metaphors.

2.2 Traditional DJ Setup

The first setup trend that is currently used by the DJ community has its roots in the dawn of the DJing, and is therefore denoted the Traditional Setup [3]. Although historically analogue [15; 16; 17; 18; 19] (no digital conversion along the signal path), it has been recently enhanced with a few digital components (namely CD players and digital effect processors), but its shape has remained untouched. Components of a Traditional setup are usually quite heavy hardware pieces.

The Traditional setup's gear is listed below[5] [15; 16; 17; 18; 19; 20]; the terminology used to describe each feature is drawn from technical-DJ vocabulary and is fully explained in Appendix D.

[5]Some components, such as the P.A. system (used for external amplification to the listening audience, if there is one), headphone and microphone, could also have been included in the list; however, they are not relevant for our work, so we left them out.

Figure 2.2: An example of the Traditional Setup, that DJs still use today.
(Taken from Kayintveen, Creative Commons Licenced)

DJ Mixer	*(single component, usually between 2 to 4 channels)*
Basic features	Crossfader, vertical faders, EQ section, input selectors, gain or trim knobs, panning knob, master volume control, VU meter, send/return section, headphone control, fader curve control and pre-listening [19]
Advanced features	Digital incorporated effects, touch pad control, MIDI control, on-board samplers, motor controlled faders, filters and exciters, fader-start technology, optical/digital crossfader, matrix routing on input and AD/DA expansion cards

Table 2.1: List of features found in DJ Mixers.

Turntables	*(usually two pieces or more)*
Basic features	Pitch control (commonly around ±10%), start/stop button, speed selector (33 and 45 RPM), arm and reproduction stylus, vertical and tangential weight calibration
Advanced features	Digital key correction, wider pitch control, MIDI control of turntable operation, accurate-music-interval pitch correction, reverse play operation, 78 RPM speed and motor speed adjustments

Table 2.2: List of features found in turntables.

CD Players	*(usually two pieces or more)*
Basic features	Wide pitch control, digital key correction, loop functions, tempo-lock, track search (rewind and forward) and hot-cue
Advanced features	Digital scratch emulation, moving platter (resembling a real turntable), built-in effects processors, cache memory, play two audio tracks simultaneously of the same CD, support for various storage media, reverse play, waveform display, MIDI control, sample recording and triggering

Table 2.3: List of features found in CD players.

Records	*(recorded music media)*
Common media	Vinyl (usually 12” but also 7”) or CDs
Advanced media	SD, HDs for the CD players and other miscellaneous storage media.

Table 2.4: List of features found in records and other media.

Figure 2.3: DJ CD-player (Technics Sl-DZ1200) with scratch emulation.

2.2.1 Sound Sources: Turntable and CD players

The turntable is the foundation of the DJing activity since the introduction, in the 70's, of the Technics SL-1200 [21], a turntable model by Panasonic that unexpectedly brought a revolution to DJing and is still the standard to most DJs. Recently, new features were added to the classical turntable, moving it a step further; included are digital processing (such as the previously mentioned key correction) and enhanced external connectivity, in the form of a MIDI-capable device.

Today, DJs accept CD players as suitable replacements for the traditional turntable not only because they are capable of playing digital records, but also because they minimize the weight of the vinyl crates; and, furthermore, recent models are capable of scratch emulation, thus creating the concept of a full digital turntable.

Scratch emulation, where a rotative jog wheel was added to emulate the turntable platter and give a more natural feeling to the end user, is a byproduct of the evolution of DJ CD players, where Denon S5000 [22] was the first to include it, and the current market leader is the Pioneer CDJ1000 [23].

Major brand manufacturers now produce turntable-like CD-players, such as the one shown on Figure 2.3; Technics designed the SL-DZ1200 [22] with a classic turntable style and Denon's S5000 model introduced novel playout capabilities (e.g., playing two tracks from same media disc simultaneously). Other features extend classical turntable mechanisms, such as waveform visualization, instant cue-point, looping, pitch bending and key lock (stretches a song without creating a musical pitch difference).

2.2.2 DJ mixer evolution

The mixer is one of the core components that made DJing, as we see it today, possible; and, starting with the first model designed by Rudolph Thomas Bozak (and still cloned by major hardware corporations [16]), has since then evolved, and new features were added: the crossfader (introduced in a model called the GLI PMC9000 [16; 20]), audio Equalizers, headphone output, and so forth.

Figure 2.4: Grand Master Flash showing his scratching skills in the 80's.
(Taken from [12])

Its fair to say that the DJ mixer plays a great role in the DJ culture [16]; where initially it was just offering the possibility of mixing two sound sources together - thus creating a "mix", nowadays it is the gear that gives more musical expression to the DJ, offering an extensive lexicon of features that DJs translate into physical gestures[6].

2.2.3 Setup Layout

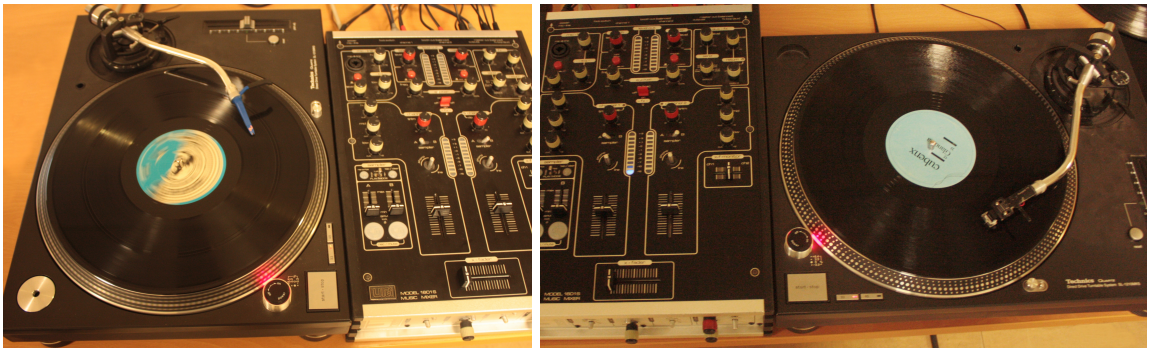
Early DJs defined a setup layout "mixer and two turntables on opposite sides" that has remained a classic for about 40 years, as shown in Figures 2.2 and 2.4. Later on, DJs started to customize the placement of the components according to their needs, with the sole objective of maximizing task performance. Hence, Scratch-DJs place the turntables vertically, as shown in Figure 2.5(a), minimizing body movements when beat-juggling and ensuring more record space to execute scratch movements without touching the tonearm or needle. Conversely, Club and Radio-DJs use turntables horizontally, as shown in Figure 2.5(b), making it easier to access the pitch slider without risking any accidental collision with the tonearm.

Although the typical DJ setup now comprises more gear than just turntables, these new components are spread around and do not interfere with the usual topologies shown on Figure 2.5.

2.3 DJ Performance Analysis

Analysing the DJ performance is critical to the process of requirements definition, which will play an important role in interface design of our approach. In order to draw a suitable analysis we used two different approaches: we researched DJ literature, and we gathered information from a DJ session with our accompanying group of experts.

[6]Turntablists are the DJs responsible for the latest evolutions of the mixer [20] and gestures performed on the cross-fader [24], as seen before.



(a) Scratch-DJ Layout (b) Club-DJ Layout

Figure 2.5: Different turntable layouts

2.3.1 DJ Research and Literature

Several academic publications target DJ analysis. Hansen et al. and Lippit developed a series of works studying the DJ performance thoroughly, with a focus on scratching activity [13; 24; 8; 14], while Jeffs and Cross focused on the hardware evolutions [20; 16] and Andersen and Beamish discussed new interface alternatives for the Traditional setup [19; 15]. Bell raises several interesting questions regarding DJ live sessions, such as the impact of new setups and improvisation within the DJ context [5]. Finally, Farrugia and Gates et al. researched the overall activity involved in DJing [25; 26].

On DJ-specific literature such as "How To DJ Right" [27], "Hip Hop Files" [6] and "Last Night a DJ Saved my Life" [2] we find valuable studies and documents of modern DJ performance in a detailed DJ terminology. DJ performances are also available on documentaries and video-recorded sets such as "Scratch" [12], an in-depth vision of turntablism, focusing the DJ activity in detail.

2.3.2 Accompanying DJ group

Throughout the development of our work we discussed several issues with an accompanying group of DJ experts. This group of five DJs, of different styles and musical genres, included two semi/professional Scratch-DJs, two professional Club-DJs and one Radio DJ that plays a weekly broadcast DJ show. As we will see through this document, the interaction with these DJs allowed us to validate requirements and study the impact of our approach.

For DJ performance analysis we conducted a typical DJing session with a Traditional setup, that was videotaped and audio-recorded (with the consent of all DJs). Figure 2.6 shows the session environment where the DJs were asked to mix a 20 minute live set with their own musical selection. The session was followed by a discussion where each DJ was asked, based on his own DJ-style and expectations, to list the

Figure 2.6: DJ expert from accompanying group in a live mix.

most relevant requirements of the Traditional setup, as depicted in Figure 2.7. In the following section we present and discuss upon these requirements.

Figure 2.7: Session with accompanying group of DJs, discussing the requirements of DJing.

2.4 DJ Requirements

We conclude with a list of requirements, drawn from performance analysis of section 2.3 and validated during the group interview with our accompanying group of DJs. The requirements of Traditional DJing are split into different categories, each directly related to actions executed on components of the aforementioned setup.

The **DJ mixer** is the center of the DJing setup. It allows the blending of two or more sound sources; several actions can be applied in the mixer sub-components, as shown on Table 2.5, thus they stand as requirements that any DJ application must fulfil.

DJ Mixer	
Fader movement	Gestures applied to moving sliding-faders, mainly for channel volume
Knob movement	Related to altering parameters of the mix, is a popular gesture in DJ-gear to interact with Equalizers
Crossfader techniques	Comprised of crossfader moves, either very fast or slow, depending on the targeted result, mainly used only by Scratch-DJs

Table 2.5: Requirements of the DJ Mixer

Sound sources are the most relevant component in any setup; usually, traditional DJs use a turntable or CD-player for sound output, so the selected device must fill some requirements that are shown on Table 2.6.

Sound source	
Audio Playout	Devices are capable of various tasks, from playing the audio content, to pausing and restarting from that point (called cue point) and forward/rewind selection
Audio Manipulation/Scratching	denotes a more advanced feature, only possible in turntables and high-end CD-players. This enables the media to be played back and forth at different speeds; and speed up/down by applying hand motions or finger pressure on the platter surface

Table 2.6: Requirements of the sound sources.

The next set of requirements refer to some manual tasks that DJs do around their setup, thus are denoted as **environment operations** as shown in Table 2.7.

Environment operations	
Record Selection	Common task in any DJ setup, usually performed by searching manually over a music library either of a CD or Vinyl case
Setup re-arragmentent	Manual arrangement of hardware components to match the desired DJ layout, depending on the DJ style
Audio Visualization	Refers to eye-witnessing the state of the DJ session. It can be: determining position or structure of a song by observing the vinyl directly or waveform in CD player display; or amplitude measurement facilitated by analogue VUs or more complex frequency monitoring performed with additional gear.

Table 2.7: Requirements of the Environment operations.

Although the next set of requirements is related both to DJ mixer and Environment operations, DJs found they were better defined if segmented out of the aforementioned classifications. They are shown in Table 2.8 and relate with a set of actions that allows DJ-users to customize their **audio routing**, by

interconnecting the cables between the various hardware devices according to their preferences. Also, routing can be used to listen to a track before it reaches the main output, which is crucial in beatmatching, since it allows users to align both tracks before sending the new song to the final output.

Audio routing	
Insert Routing	Manually connecting input/output pairs of audio cables
Pre-listening	Accomplished by triggering a switch or fader in the DJ mixer.

Table 2.8: Requirements of Audio routing.

These are the requirements of typical DJ systems. The industry of DJing has used them as the basis for the development of newer digital setups, such as the ones described in the next section.

3 Related Work

We've got Two Turntables and a Microphone

Mantronix *in Needle to the Groove, 1985*

In the present section we start by looking at commercial DJ solutions that encompass the first two types of setups, Virtual and Hybrid, and are the top choices for contemporary DJs, as they enable us to study how DJs interact with digital tools. Then, we look at a wide range of academic proposals of the tangible, wearable and haptic approaches; they represent the state of the art in DJing and loop-sequencing reasearch that is significant to our proposal, and therefore are also mentioned here. Furthermore, we acknowledge the value of recent developments in musical controllers based on multitouch interaction, as their immediate success in the DJ community motivates us to perform a more in-depth research.

Also one must stress that there is a vast amount of research information on DJing across a multitude of platforms, so the scope of this survey is limited to professional DJ systems[1] that run on personal computers or similar machines, and to academic proposals that address some of our DJ requirements and concepts. Therefore, we do not address DJing systems for casual mobile-phone operation or gaming/educational purposes[2].

Finally, a critical comparison of the contributions is summed up, helping us to pave the way for the proposal of a multitouch setup for DJs.

3.1 Virtual DJ Setup

Computers enabled the use of digital tools for any art form one can imagine; from painting [28; 29] to music creation [30; 31; 32], everything can be done in a 0-1s environment, and so can DJing.

However, early Virtual DJ systems had a long journey before they convinced DJ fanatics, that they could bring their laptops to the club. In fact, only in the mid 90's and early 2000's [33], did these DJ

[1] This denotes the setups that comply with the typical DJ requirements that we gathered in section 2.4.

[2] There has been an increasing number of new DJ applications targeting mobile phones and gaming consoles, which are not representative of professional applications and only adequate for casual users.

systems start to be widely accepted by DJs, partly due to inadequate implementations that caused audio applications to run with poor response times (high latency and slow processing) and also because of non-natural [34] interface mappings or poor functionality.

As mentioned before, the Traditional setup exists for at least forty years - almost untouched in its core. Therefore, making it an "all virtual solution" can be a though challenge both to developers and users.

Virtual systems have evolved a lot in a short amount of time, and nowadays they are quite powerful. Despite lacking "real" touch of the DJ gear, they also bring along some important advantages to musical expression, ones that DJs take in serious consideration: the virtual mixer has no track limits - playing as many songs as the user intends, offers a complex way of manipulating the audio through digital signal processing, is a cheaper and portable solution with regard to traditional hardware, has an unlimited and weightless storage medium, and so forth.

In the following sections we present a set of application examples from the commercial DJ world that show relevant features of Virtual setups and their contribution to the DJ task.

3.1.1 Ableton Live

The popularity of Ableton Live [35], rises everyday, being a widely accepted and acknowledged application for computer musical performance [36], used everywhere from DJing [37] to the more experimental realtime electroacustic context.

What makes Live stand out from its famous competitors [3] is directly related to simplicity, rather than offering something new. Live sparks in terms of ease of use and low-learning curve, featuring an award-winning interface design [38], shown on Figure 3.1. As far as audio manipulation is concerned, Live offers an automatic timestretch tool (preserving a sample's musical pitch, but expanding it in time, thus changing the song's speed) and automatic quantized and in-*tempo* sample launch.

As far as extensibility goes, Live accepts VST plugins [38] and can be rewired[4] with other programs. With a hand full of features and simplistic approach, it is quickly becoming the number one loop-based program [40].

Although Live is not targeted to DJing, there's a trend in using it to mix tracks easily, due to the aforementioned time-stretching capabilities, that facilitates the beatmatching job for the DJ. Because Live offers the possibility of two work modes, Session View (for interacting with loops/samples) and

[3] The list includes realtime-loop-based applications such as ReNoise, AudioMulch, Qtractor, Hydrogen, Reason, FL Studio and others.

[4] Rewire is a software protocol by Propellerheads and Steinberg to allow multi-track MIDI and audio connections between audio applications [39].

(a) Session View (b) Arrangement View

Figure 3.1: Live is flexible for realtime performance, by offering two distinct views. [35]

Arrangement View (for pre-recorded performances), DJs can easily use it to play and manipulate their songs.

In-depth papers analysing Live have been published by Zadel and Scavone [41], Blackwell et al. [38] and Cronin [36]. According to these authors, the relevant features that made Ableton Live a success to Virtual DJing are:

Simple Tool Live's features are a small limited list, thus making it a consistent and easy-to-use application. It is also attractive to DJs due to automatic time-stretching of songs.

Clean Interface The graphical aspect of Live is the key to its simplicity: navigation and selections are simple tasks, and there are almost no message boxes or sub-menus.

Mixer look-alike Although Live is not a direct graphic representation of the real gear, its key widgets are plain faders and knobs, making the user comfortable and being really easy to learn.

External Control Live can be controlled via simple mappings (keyboard, MIDI, OSC, and Max/MSP direct connectivity [35]).

User centered design Throughout the development of the application user input was constantly taken in consideration; furthermore, Live was co-produced by musicians, such as Robert Henke[5] [37].

A final remark is that Live [35] does not offer built-in support for connection with real turntables for digital control, a trend that the majority of DJ applications are following and one that will be analysed in Hybrid setups (refer to section 3.2).

3.1.2 Virtual DJing Applications

Several applications could be considered for this survey, all homogeneous in terms of features and target audience. We chose djay[6] for this case-study, but nevertheless evaluated some of its competition

[5]Henke's musical persona is Monolake, a project with highly acknowledged contributions to electronic music developments. `http://www.monolake.de`, accessed on 10/11/2009

[6]`http://djay-software.com`, accessed on 10/11/2009

as well, such as VirtualDJ[7], DjDecks[8], TraktorDJ[9] and, more recently, FlyLoops[10].

djay is a typical Virtual Djing system, offering the same features as other already cited software applications[11], all operated on the basis of a strong dual turntable metaphor. The user interacts with the turntable widgets to control the playout of songs, mixing them with virtual faders on a mixer-like widget.

(a) djay (b) Traktor DJ (c) Virtual DJ

Figure 3.2: Screenshots of three Virtual systems
(Taken from [6,7,9])

We conducted a survey on the aforementioned applications, and gathered a feature list; similarities between them are remarkable, and show what a typical Virtual DJing application can do:

Traditional Setup Look. To make interfacing easier, the virtual controls resemble Traditional DJ gear setup (Figure 3.2).

Audio manipulation Typical features of audio manipulation are pitch-change (varying from ±4% up to 100%), time-stretch, play/stop/pause, EQ, Sampler/Looping widget and external effects integration.

DJ Aid tools. Fast and accurate automatic BPM detection and automatic mixing mechanisms (auto crossfader transitions and automatic BPM-adjust) and visualization tools (spectrum analyser and stereo VU-meter).

Audio Routing. Standard among Virtual DJing, featuring pre-listening on each track.

Scratch Emulation. Some Virtual DJing applications feature scratching emulation. For the case-study of djay, it allows users to interact with the turntable widget via mouse input or multitouch gestures performed on laptop's touchpad. They also account for other features such as shortcut keys for crossfader motion and brake/start-up speed of the turntable emulation. However, as will be seen in the discussion, these emulations are widely criticised by users due to the lack of natural interaction that the mouse-device offers.

[7] `http://www.virtualdj.com`, accessed on 12/11/2009
[8] `http://www.djdecks.be`, accessed on 11/11/2009
[9] `http://www.native-instruments.com`, accessed on 03/11/2009
[10] `http://www.flyloops.com`, accessed on 27/12/2009
[11] Only exception is that it integrates with iTunes music library directly.

Figure 3.3: A MIDI DJ controller coupled with a Virtual DJing application.
(Taken from Oobrien, Creative Commons Licenced)

This type of application is targeted at mobile-DJ users[12] that do not intend to carry extra heavy-controllers or complex Hybrid systems; they usually accept external MIDI control and spread MIDI-DJ controllers, such as the ones in Figure 3.3, rendering them quite popular. Therefore, Virtual DJing applications are frequently paired up with this type of controllers.

3.1.3 Discussion on Virtual DJ setups

Virtual DJ setups flourished in the turning of the century and raised the debate over digital DJing. As an example, Farrugia et al. [25] and Gates et al. [26] papers and articles by the NY Times magazine [33], both analyse digital DJing in terms of pros and cons.

On the plus side, the benefits from full virtualization (the entire setup is digital within the computer) include using digital songs (no records to carry, no record wear out, and simple collection management) and minimizing transportation and maintenance costs. However, negative user feedback comes specially from DJs that prefer the Traditional setup, such as vinyl fans and turntablists that require real turntable operation (these experts of scratching do not settle for digital emulation as offered by those software applications mentioned above).

3.1.4 Virtual towards Vynil Tracking

One additional interesting aspect about the Virtual setup, is that the manufacturers are coupling them with Vinyl Tracking software, allowing users to control digital audio files via a regular analogue turntable coupled to an analogue-to-digital converter unit (covered in the next section). This enhances the Virtual software (it becomes a part of a Traditional setup that has real turntables) to control digital audio files - turning it into a Hybrid setup (Figure 3.4).

[12] A note for the reader: mobile-DJs are the segment of the DJ taxonomy that uses mainly Virtual systems for maximum portability. They perform in clubs and events on a daily basis and travel constantly. Thus, the word "mobile" in this context does not refer to mobile-phone systems or embedded technologies. (see Appendix B for DJ taxonomy)

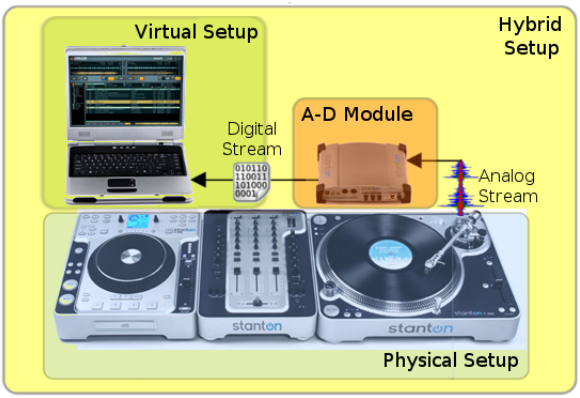


Figure 3.4: Relationship between the three setups.

In the applications previously studied, there are two cases of recent additions to vinyl control: one is DjDecks and the other is the partnership of Native Instruments (NI) and Stanton Magnetics[13] to allow vinyl control in the 2003 version of Traktor.

3.2 Hybrid DJ Setup

The Hybrid setup is currently taking the DJ world by storm; as the name implies it merges the features (and advantages) of the Traditional and Virtual, allowing a more physical interaction between the user and the Virtual setup, as shown in Figure 3.5. Applications that feature vinyl control are follow-ups of Virtual systems, therefore maintaining a dual turntable metaphor; they do not extend the feature list of the Virtual software, except for the obvious fact that they allow turntable/scratching emulation.

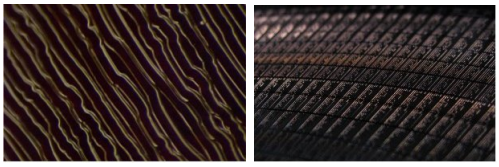
Almost every Hybrid setup is composed of a Virtual system (that ultimately runs on a computer's core) and a set of physical controls - Figure 3.4. Classifying a system as Hybrid may be misleading, because DJs may use full Virtual systems with accessory expression controllers (usually in the popular MIDI protocol format [42]) to control the application, without using computer input devices (typically mouse/keyboard). Our definition of Hybrid system embodies a setup that has at least one component found in the Traditional setup and one Virtual system; this categorization of the "Hybrid DJ setup" is also proposed by Bell [5].

Nowadays all vinyl emulation applications adhere to the current technological trend: time-coded vinyl tracking. Although Final Scratch (shown on Figure 3.5) was probably the first to appear and gain notorious use among DJs, competition soon followed, and now it offers the same set of features.

[13] `http://www.stantondj.com`, accessed on 28/10/2009

Figure 3.5: The Final Scratch system, being controlled with turntables.
(Taken from [13])

Vinyl tracking works using special vinyl records, recorded with a time-code signal, as exhibited in Figure 3.6(b). The computer's software interprets this signal (a digital signal, after being converted by the A-D module in Figure 3.2) and determines the position of the needle on the record, what direction the arm is moving, and the its speed; these calculations give precise parameters on the turntable's state at each instant: time offset in the song, and motion values (forward/backward and acceleration). This information is mapped to the currently selected digital audio file, and processing emulates a classic turntable behaviour on top of the audio stream.



(a) Analogue Audio (b) Digital Time-code

Figure 3.6: Macro shots of (a) analogue audio and (b) timecoded signal on vinyl.
(Taken from Jason Ruck, Creative Commons Licenced)

DJs became really fond of the Hybrid setups' advantages, allowing them to travel with thousands of songs on a small laptop and still use the same techniques they mastered on their classic DJ gear (turntables, CD-players and mixer).

The Hybrid DJ setup has been analysed in academic papers and surveys [43; 23; 22; 44; 45; 17; 46] and also by competing products - either similar ones such as the Mixxx application of Andersen [47; 19] or new multimodality approaches that take Hybrid setups as a starting point [14; 15]. There are also reports that focus on the DJ-user's acceptance/reluctance to Hybrids [25; 48; 49].

The best Hybrid case-study in the industry is probably Final Scratch, a software/hardware package

Figure 3.7: Final Scratch uses the dual turntable metaphor and hardware for timecode tracking. (Taken from [13])

initially designed by Dutch company N2IT with direct input from DJs Ritchie Hawtin[14] [50] and John Acquaviva. Later on, it was merged with a Stanton Magnetics hardware interface and with NI's Virtual system Traktor Pro software. Although they currently dismissed the partnership[15][16], it is still by far the most innovative vinyl tracking system that appeared in the DJ market.

True innovation came from the natural interaction that Final Scratch allowed [15]: although users do not actually manipulate the sound content of a vinyl record, they use the Traditional setup in a natural way (as described in section 2).

Nowadays DJs are widely accepting these Hybrid systems as they perform faster and without noticeable latency; the competitor's list features many systems[17]. Nevertheless, our focus will reside on the following two, because they offer unique aspects over the others:

MsPinky [52] is a very interesting case study, although it suits programmers better than regular-users due to its low-level interface for customization. First, because it strives to fill a gap in cheap Hybrid systems[18] (thus eliminating one of their biggest disadvantages: high pricing) and secondly as it encourages users to develop their own solutions on top of the MsPinky SDK. MsPinky includes two standalone applications, Maxi-Patch(Windows or Mac) and Binky Toy (Mac only), a VST plugin called Pinky VST[19] and a Software Developer's Kit (cross-platform) which allows programmers to access MsPinky's Vinyl Tracking Object (MPVT) in their own code, as shown in Figure 3.8.

[14] Hawtin is best known for his alter-ego Plastikman, an acclaimed electronic-music producer.

[15] Stanton changed it to Final Scratch Open and now distributes the hardware for usage with other applications that support their timecode [51],

[16] Traktor Scratch is a remake of Traktor Pro after the break-up of NI and Stanton, ending the Final Scratch project. Today, NI has their own hardware interface and has released the new application to the market.

[17] Serato Scratch, Mixvibes, Torq, TouchDVS, PCDJ Scratch, D-Vinyl 20-20, Quad (all commercial), and terminatorX (Linux-only) are quite similar applications for vinyl-control, all based on timecode and scratch emulation. As they do not offer any distinguishing features, we won't consider them for further analysis.

[18] Currently costing 99 dollars [52].

[19] This is a striking advantage to DJs, as they do not need to change working environment/application to move to a vinyl-controlled situation.

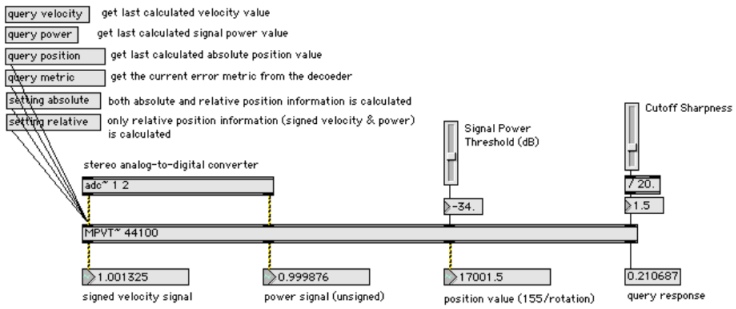
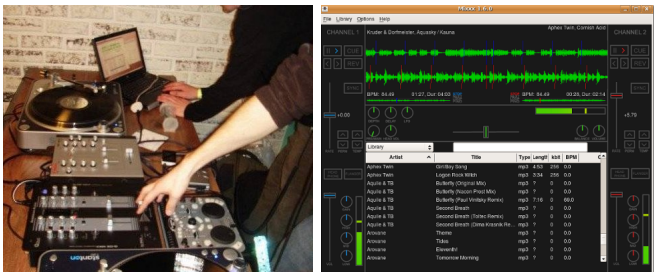


Figure 3.8: A programmer receiving timecode interpreted data within Max/MSP environment. (Taken from [52])

Mixxx is an open source alternative to the previous systems[20]. The project was started in early 2001 [47] (when only a few systems where active, Final Scratch being one of them) and was created with quite a lot of input of various DJs. It offers the same functionality as its competitors, and includes a few DJ-aid features such as beat estimation, and parallel visual displays.

For scratch emulation, it uses ScratchLib[21] and xwax[33] to decode various popular timecoded vinyl formats. Mixxx creators state that latencies of 10ms or less can be achieve if powerful CPUs are used, thus proving it adequate to DJing requirements and worth to be included in the list, along with commercial products [53].



(a) Mixxx with turntables (b) the Interface

Figure 3.9: Mixxx being used in a performance-situation. [53])

Oddly as it may sound, Mixxx has features that make it remarkably stand out among competitors: multiple soundcard support, cross-platform distribution, advanced MIDI scripting engine[22] and Multi-core CPU support.

[20] An interesting side note: Mixxx is currently included with the Ubuntu Studio distribution, an open-source operating system targeted at multimedia production. `https://help.ubuntu.com/community/UbuntuStudio/Applications`, accessed on 29/12/2009

[21] ScratchLib and xwax are open-source applications for timecode signal interpretation.

[22] With its scripting ability Mixxx can internally create, at runtime, new MIDI mappings from trigger inputs, thus being able to create complex MIDI routes for the more demanding DJs [54].

3.2.1 Discussion on Hybrid Setups

The Hybrid applications analysed here [29], although widely accepted for their advantages, do have some disadvantages either over Traditional setups, based on conventional analogue records, or over Virtual setups. Based on the quoted papers, our conclusions about the Hybrid advantages are:

- Portability of thousands of audio tracks, as opposed to the size and weight of record boxes carried with Traditional setups. Digital media also offers quick-access and easy organization of the music library.
- Ability to play and manipulate audio tracks that are difficult to get in vinyl (such as rare tracks, remixes or even self recorded material). Also aids in preserving records, as they can be easily backed-up.
- Availability of CD player features: keylock, pitch shift, looping, instant cue and visual indicators of audio track contents as Dj aid tools.
- Some software implementations prevent needle skipping events on vinyl to be reflected in the audio playback, a quite helpfull feature for the DJ performance.
- Integrates seamlessly with Traditional DJ gear and allows users to keep on performing the same performance gestures. Extension of DJing expressiveness is achieved through capabilities in digital audio manipulation.

On the other hand, Hybrids have certain disadvantages, namely:

- Reliability can be a serious issue here, as Traditional setups are known to be bullet proof systems - with rare faults, while Hybrid setups depend heavily on the hardware/software configuration used. Vinyl emulation and signal decoding can use more system resources than some laptops or PCs offer, therefore making the whole system unstable which, for professional usage, is a serious disadvantage.
- Most Hybrid setups only allow two tracks to be played simultaneously, thus loosing one of the biggest advantages of Virtual setups.
- Closed hardware/software can be an hindrance; although systems such as Mixxx, TerminatorX and MsPinky are open for developers, the current popular timecode vinyls had to be hacked to fit these systems. Dependence on closed hardware can also be an issue for application packages like Final Scratch or Serato Scratch[29] (they only work with their on, certified components). Also, software dependencies can cause problems with lack of support - as happened with Linux support for Final Scratch versions being dropped [23] or with the break-up between Stanton and NI, ceasing the distribution of their "joint product".

[23] Online Petition from Final Scratch Users. `http://www.petitiononline.com/PCMACXP/petition.html`, accessed on 21/12/2009

Figure 3.10: AudioPad interface in a collaborative session. [56])

In conclusion, Hybrid setups based on vinyl tracking solve some of the issues of Virtual applications, giving the DJ a more natural and tangible tool to control the audio and, additionally, offering digital audio storage and digital processing of the audio data. But the system will not be as cheap as a full Virtual one, will be more limited in terms of simultaneous tracks and features, hardware component freedom of choice, and will also inherit the Traditional disadvantages (mainly costs of transportation and maintenance).

3.3 Tangible Approach

Previously mentioned DJ setups have new competitors; for example, the new trend of multi-modal technology interfaces. Although these interfaces offer music creation possibilities, they are not targeted specifically towards DJing, but implementations such as AudioPad and Reactable show new paradigms and possibilities regarding tangible and touch manipulation.

Four of the most representative tangible interfaces for musical composition were chosen: AudioPad, BlockJam, Reactable and Pin & Play & Perform, all conceptually designed around the same "building-blocks" metaphor, well publicised in the literature on tangible interaction [55].

3.3.1 AudioPad

Patten et al [56] propose AudioPad, a modular musical interface designed for tangible interaction on tabletops. The system consists on a set of tangible graspable objects that users position on the surface to control various aspects of the live-musical performance, e.g., to trigger musical loops and sequences, control effects, mix samples, and so forth.

The project uses RFID tracking, as previously proposed by Pasadiso et al. [57; 58]. Tracked objects are communicated to the system, which ultimately controls the musical output, and displays and the user's in-table interface visualizations, as depicted in Figure 3.10.

The natural user interface created within the project is a very interesting case-study (featuring floating-menus and advanced selection mechanisms) and, in [59], the authors describe Audiopad's interface concepts and design guidelines thoroughly.

3.3.2 BlockJam

BlockJam by Newton-Dunn et al. [60; 61] is a musical interface that provides tangible interaction between users and magnetic-coupled blocks, thus being capable of collaborative musical composition in realtime. The user reorganises the blocks in a desired layout and influences musical arrangement and phrasing through the different possible connections he/she has laid, while visual feedback on the blocks' state is shown on each facet (refer to Figure 3.11(a)).

(a) Interacting (b) Layout of blocks

Figure 3.11: Blockjam in the musical process (a), a layout of blocks (b). [60])

Acting on blocks changes their functionality (audio manipulation behaviour); this leads to complex and engaging musical configurations that can be triggered quite rapidly, all made possible by this simple physical interaction schema, which is a far more simple/natural metaphor than the other tangible applications featured here.

3.3.3 Reactable

The Reactable was proposed by Jordà et al. as an interface [62; 63] for musical collaborative expression in realtime. In the Reactable a set of players can share control over the instrument by touching[24], rotating and moving physical objects on the table surface to build different audio networks, an approach that resembles a flow-controlled graphical programming language[25]. Each Reactable object represents a different audio concept, or component, with a dedicated function: generation, modification or sound control.

[24] The Reactable objects also receive input parameters by touching the surface around them, because graphical widgets of those parameters are rendered onto the visualization surface [63]

[25] A suitable example is a patch representation of a Max/MSP or Pure Data program [64].

(a) Collaborating (b) Object Behaviour

Figure 3.12: (a) Reactable in collaborative session and (b) the diagram of object behaviour. [63])

The project implements a proximity concept in object interaction and, therefore, certain objects affect others when entering their proximity radius. The surface-visualization allows user to acknowledge each object's behaviour (refer to Figure 3.12(b)), their parameter values and configuration states.

The Reactable combines playing and editing (establishing connections, altering parameters) in real-time and in a unique way [65], creating a user-friendly, seamlessly integrated musical creation environment. Although early versions were too synthesiser-based[26], it now has sample-based features.

3.3.3.1 Scratch Implementation on the Reactable

Hansen and Alonso further enhanced the Reactable with an implementation of a scratch-object [66; 67]. Their scratching emulation makes the Reactable an attractive system for our own proposal, which is based on Hansen's Skipproof software [24; 68].

The turntable metaphor was added to the Reactable and uses tangible objects for scratch control. The Reactable DJ-user has three different objects to operate on: the loop-player (plays back audio and represents tracks progression), the movement-speed object (controls turntable motion and speed) and finally the cross-fader object (that triggers a crossfader movement inside Skipproof); with a combination of the three, a user can achieve the typical sound of scratching. As usual in the Reactable behaviour, objects communicate through lines (indicating the connections and audio streams) and the scratch volume can be controlled by moving the sound source closer to (or away from) the output's center point (refer to Figure 3.13).

Hansen and Alonso had a handful of users evaluating scratch objects [67]; results they gathered are important to understand how tangible interactions, such as our proposal, can be used in a DJing application:

- The Reactable-expert user became more optimistic, with the setup, as the experience progressed

[26] The Reactable core is made of a Modular Synthesis and Dynamic Patching approach [65], creating several software synthesisers that could be patched and altered by the tangible manipulation.

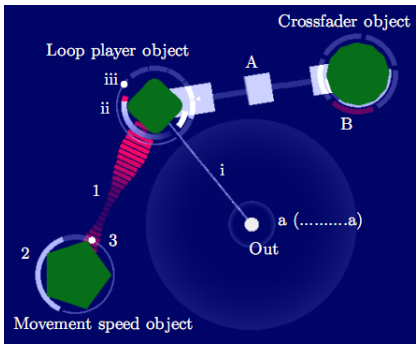
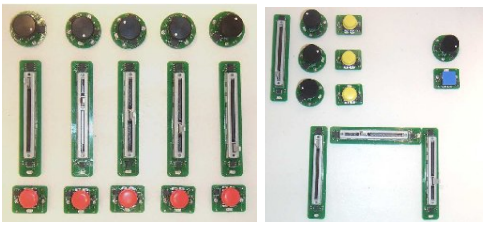


Figure 3.13: Scratch objects in the Reactable environment. [67])

and gained a deeper understanding on how to relate with the Reactable behaviour and how to trigger the modelled scratch effects.

- The DJ-user felt more pessimistic about the system as the test progressed, showing increasing discomfort with the control objects. They also identified that these systems cannot match the scratch experts' expectations of analogue turntable behaviour.

3.3.4 Pin & Play & Perform



(a) Club DJ (b) Scratch DJ

Figure 3.14: Different states of the Pin & Play & Perform interface arrangement. [69]

With Pin & Play & Perform [69] (PPP), a tangible, rearrangeable interface is proposed. The framework includes a set of DJ objects such as potentiometers, triggers, dials, and so forth, that can be rearranged on the fly, allowing the user to compose an interface that suits his needs. The prototype is exhibited in Figure 3.14, where one can create interface arrangements to suit both Scratch DJs (Figure 3.14(b) shows a crossfader based arrangement) or Club/Radio DJs (Figure 3.14(a) where the interface is heavily populated by vertical, rather than horizontal, faders).

PPP is targeted at providing a flexible MIDI controller with customizable layout, where new components can easily be added, showing how important interface rearrangement is for professional DJs.

Figure 3.15: User interacting indirectly with a Virtual application using PPP. [67])

However, it does not offer on-surface visualization as displayed in Figure 3.15, therefore inheriting the disadvantages of indirect manipulation [70].

3.3.5 Discussion on Tangible approaches

One must remember that projects we have covered in this sub-section have different objectives, rather than being typical DJ systems. In the first set (AudioPad, BlockJam, and Reactable) their authors agreed on a novel metaphor for music sequencing. Furthermore, their focus is on studying the collaborative process of musical creation, a concept more suited to musical composition than to standard DJing. As for the last one, Pin & Play & Perform, its author mapped traditional DJ concepts to a rearrangeable interface, allowing us to witness the importance of layout to professional DJing, thus marking it as a mandatory requirement in our own project.

As far as tangible objects on tabletop surfaces are concerned, we see that Skipproof embedded in the Reactable showed interesting results. However, it does not seem to offer any extraordinary features that would make it stand out as a DJ controller with turntable-like emulation, mainly because it does not map traditional gestures easily.

Closing up on our tangible-object interfaces review, we acknowledge their potential for hands-on audio applications, as they encourage two-handed interaction and an external visual representation of the Virtual system [55]. They are currently well suited for serious loop-based composition and realtime modular synthesis[27], but lack DJ-specific features and its traditional lexicon.

[27] A remarkable user of Reactable is musician Bjork, in her live performances. `http://www.reactable.com/reactable-systems-is-ready-to-go/`, accessed on 21/12/2009

Figure 3.16: Control scratch movements with sensors. [71]

3.4 Wearable approach

The next proposals show how wearable hand-controllers can suit DJs in realtime performance. Being heavily sensor-based gives precise gestural information that can be mapped into coherent gesture recognition. However, we see that while these interfaces may exhibit good record control, they lack other features and gestures and, moreover, they force users to wear complex glove-like devices that limit the typical hands-free scenario of traditional DJing.

3.4.1 Sensor-Glove for Skipproof

Sensor-Glove is a haptic glove device proposed by Mandoux et al. [71], that controls the Skipproof scratch emulation program by Hansen [68], supported by a richer set of data collected at each sensor, as shown in Figure 3.16. Although it is able to map new gestures to scratching techniques, the interface is quite detached from the user's natural environment, making it difficult to use in a live-act situation - that is, without proper training.

3.4.2 DJammer

DJammer is a project sponsored by HP-Labs and developed by Slayden et al. [9; 72]; it builds on the same set of ideas used in the sensor glove, but expands its possibilities by creating a portable device, rather than just an interface.

The DJammer can trigger various scratch techniques and other typical DJ actions through data collected from various sensors (accelerometer, optical, and different types of buttons) - its internal workings are summarised in Figure 3.17. Although it is far more mature than the proposal of Mandoux, it also suffers from the same drawback: it expects DJs to learn a new vocabulary of gestures. But anyway, tests performed on Djammer showed that altough it is suited for portable operation, it is not adequate to support the entire performance in a DJ booth.

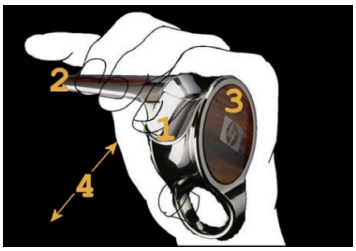


Figure 3.17: The mapping of Djammer: (1) for crossfader control, (2) to place a hand on a record, (3) to lower the needle on the record and (4) to move the record back/forth. [72]

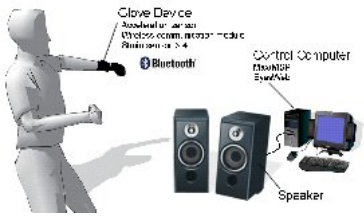


Figure 3.18: MusicGlove system architecture and components. [73]

3.4.3 MusicGlove

Hayafuchi et al. [73] devised a wearable musical interface, as depicted in Figure 3.18, which could also be used for DJing. By combining information from several pressure and acceleration sensors, they are able to provide accurate hand-gesture recognition, which is then mapped into DJ actions. It offers all the typical features: scratch, fast-forward, rewind, play, stop, and so forth. Although MusicGlove behaves naturally when compared to physical turntable manipulations, it uses gestural mappings that are nontrivial to a traditional DJ; as an example, scratching motion is only recognised if performed without any vertical movement of the hand, something that is quite difficult to control when performed "on the air" - scratch is typically performed on a record surface, which guides the user hand on an horizontal plane.

3.4.4 Discussion on Wearable approaches/devices

While some of these glove-like solutions can provide natural gestures as far as DJing is concerned (and therefore suitable metaphors to accurate precision scratching), interaction does not lack problems, such as the behaviour in resting positions (i.e., when DJs lower their hands to interact with other components of the setup) or hands-free interaction, which is simply not possible. Furthermore, being based solely on "air gestures", they provide no tangible/haptic feedback on movements. Ultimately these gloves, if used without force-actuators [74], will provide worst sensory feedback than haptic, tangible or even multitouch

tabletops.

3.5 Haptic approach

Haptic devices in the DJ context are physical approaches to turntable behaviour, where tactile feel is usually mimicked by force feedback to emulate the rotating platters. We overview two similar proposals which target DJs that need good tangible feedback, such as Scratch users.

3.5.1 D'Groove

D'Groove is a Hybrid haptic setup for digital audio control proposed by Beamish et al. [75; 15]. Its interface is used to manipulate digital audio, and it does not need real turntables or time-coded vinyl.

The remarkable advantage of D'Groove is the natural interface it offers to end-users, which do not need to learn new gestures [15]. The system's components are the haptic turntable, the pitch slider and the queue slider, as shown in Figure 3.19; this set emulates both a normal turntable and a mixer for DJing.

Figure 3.19: D'groove system with the three different components. [15])

Tests conducted by the authors showed that D'Groove can perform quite well, even when used by scratch experts, thus making it a capable DJ controller. It also provides DJs with a set of tools for beat-matching, because it uses haptic feedback to send "bumps" (vibrations) to the user to signal the song's beats. Visual markers placed on the vinyl serve as visual cues for the DJ techniques (specially scratching, beat-juggling and beat-matching).

3.5.2 ColorDex

ColorDex [76] can be categorized in two axis, as it allows direct haptic manipulation of the audio via a turntable-like device (HDDJ, Figure 3.21) and also a novel tangible manipulator (Cubic Crossfader, Figure 3.20) that enables gestural mixing of up to six audio tracks. Although the device has not been

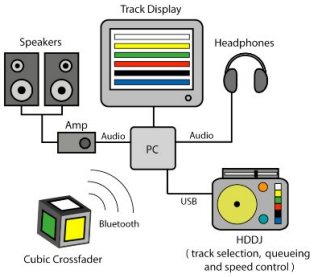


Figure 3.20: Overview of ColorDex components, including a tangible crossfader and a haptic device. [76]

Figure 3.21: Detailed view of HDDJ haptic turntable. [76]

extensively tested in real DJ situations, preliminary tests showed that both direct manipulation and tangible mixing are adequate for basic DJ tasks.

3.5.3 Discussion on Haptic approaches

While studying papers that cover haptic systems we noticed that they tend to focus on the interaction possibilities of scratching and discard the remaining requirements of DJing; they are not, therefore, complete solutions to our problem.

Haptic controllers adequately solve the complex mappings problem; as an example, the D'Groove proposes a natural DJ metaphor: the turntable. Although D'Groove user test results are interesting and validate the proposal [15], they have limited possibilities and they inherit the Hybrid disadvantages: they are expensive, damageable and are not easily re-arrangeable or upgraded. Finally, ColorDex also requires the DJ to learn a new lexicon to manipulate the tangible objects.

3.6 Multitouch Controllers

Musical controllers have been around for quite a while, mainly due to the pervasiveness of the MIDI protocol - a "de facto" standard for audio control - but also because they provide a more natural interaction method for musical-related tasks. While they have evolved a lot since the first proposals of Don Buchla and Robert Moog [32], touch-based controllers have only been accepted amongst DJ users recently, due to the popularity of the Lemur (for reasons that we analyse in this section). Not including the commercial ecosystem, there are other solutions that propose new interaction metaphors towards traditional DJing.

3.6.1 Lemur

Ever since its first public presentation in 2004, at IRCAM, JazzMutant's Lemur has been widely adopted by DJs as a capable all-around controller [77] and considered a major innovation when compared to the traditional static-MIDI controllers.

Lemur [77] can be used to perform some digital DJing tasks; like all controllers in this subsection, Lemur is multitouch-based, but then it is only an external controller[28], delivering MIDI messages to an application.

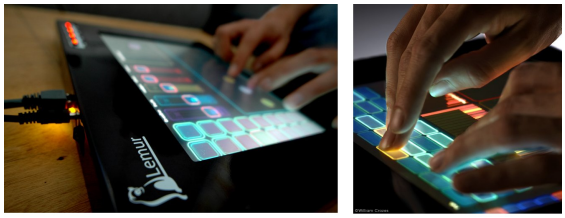


Figure 3.22: The Lemur device in touch control. [78]

The main differences between Lemur and typical MIDI controllers are:

Multitouch Lemur has no physical controls (i.e., knobs, faders, IR beams) and everything is touch-screen based; all operations are performed on the touch-screen surface (Figure 3.22).

Modular Lemur's great triumph is in the interface metaphor: its (on-)surface interface is fully customizable by the end-user (Figure 3.23), and new controls can be added, moved, resized and mapped onto any OSC/MIDI message.

[28] Actually many of the proposals we have been looking at are, in fact, controllers from a software-architecture perspective, i.e, the prototype of AudioPad is also just a controller that runs Ableton Live "under the covers" for audio manipulation [56]. However the message we want to convey here is that Lemur targets the MIDI/OSC controller market, thus does not intends to be an audio application.

Figure 3.23: Lemur running an interface for Traktor DJ. [78]

Lemur works best when used as a touchable controller coupled to a Virtual setup; therefore, DJs use Lemur with DJing applications such as Ableton Live or Traktor Pro DJ. Lemur's interface can be entirely constructed by the musician/DJ using a WYSIWYG application specifically designed for that task; Figure 3.23 shows a user rearranging Lemur's interface for DJing - its resemblance with a typical Virtual setup is remarkable.

Lemur has been studied in papers by Kobayashi and Akamatsu [79], Crevoisier [80] and Kellum [81], Furnell et al. [82] and Davidson et al. [83]. It has also been covered in the specialized press articles such as in the Computer Music Journal [84; 85], ACM Communications [86], Cronin [36] and Sound-on-Sound in 2007 issue [87]. Having thoroughly analysed the papers and articles quoted above, our conclusions on the Lemur's adequacy towards DJing are summarised below.

Advantages: Lemur offers a wide range of objects starting with the common ones that can be found on Traditional DJ gear (knobs, faders, button pads), but also including other objects such as 2-axis controllers and graphs. More "advanced" GUI widgets are also available, such as multiple cursor canvas, virtual-trackballs and curve-drawing tools. These allow users to interact with custom physical-behaviour "balls" (i.e.: with gravity and/or friction) to control their audio performance [78].

Lemur is portable and hardware powered, i.e., it does not need CPU cycles, which are then available to run the DJ Virtual applications. Lemur's communications use the state-of-the-art OSC protocol [88] over Ethernet at very fast rates: Mb/s to Gb/s.

Although Lemur is a closed-source product, it also interfaces with popular audio software, including some of the Virtual and Hybrid systems reported in our survey.

Disadvantages: although Lemur[29] now offers new objects such as recognition-objects for pinch and rotate gestures, there is still no appropriate widget for the turntable-metaphor. Oddly enough, Lemur does not offer a rotative-control that mimics the turntable movements and would be adequate to emulate of a vinyl record, even when considering the physical emulation/engine integrated in some Lemur objects.

[29] V2.0 is currently the latest version [77]

Lemur is a highly expensive product[30], surpassing the costs of all other mentioned Hybrid/Virtual DJ systems or external MIDI controllers.

Even though one could think that Lemur offers a widely extensible architecture, its important to keep in mind that this is a closed product. There are no APIs or open possibilities of developing self-objects, changing the core-interface look or functionalities [89].

3.6.2 Attigo

Figure 3.24: Attigo prototype being used instead of turntables. [90]

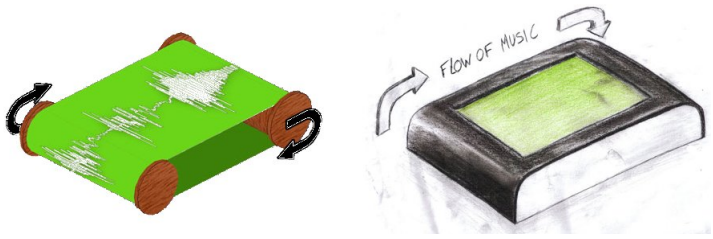


Figure 3.25: Attigo uses a new metaphor: a moving waveform in a conveyor belt-driven surface, instead of a rotating platter. [90]

Attigo is a recent project, a conceptual design of Scott Hobbs [90] thet incorporates lots of input from professional DJs. It is a virtual turntable with multitouch technology, or, if you prefer, a touch-capable device that emulates turntable behaviour. Attigo could be considered a part of a Hybrid Setup (because it requires a real mixer in the chain) but at its core it is better if one regards it as a multitouch controller, that ultimately provides motion data to a Virtual application. It also distinguishes itself from other systems we have analysed because it proposes a "conveyor belt" interface, illustrated on Figure 3.25, to replace the typical "revolving platter".

The developer gave a series of interviews to the specialised press[31] and has publish both studies

[30] Currently costing about 1800-2000 euros [87; 77]
[31] `http://www.gigacrate.com/Articles/?p=343`, accessed on 01/12/2009

and progress reports on his personal webpage [90]. The current Attigo prototype, shown in Figure 3.24, roughly has the same capabilities of a typical turntable (12% wide pitch control, start/stop, reverse, power off, and scratch behaviour).

3.6.3 Stanton SCS.3d and 3m

(a) SCS.3d (b) SCS.3m

Figure 3.26: Stanton SC System® 3, a touch-based MIDI controller. [3]

Both SCS.3d[32] and 3m are part of the latest Stanton's "SC System® Control Surface" product line. They are compact USB MIDI controllers with several two-finger touchable areas. Figure 3.26 shows that resemblance between the SCS.3d (a) and a turntable, and between the SCS.3m (b) and a mixer is not coincidental: Stanton expects to easy the user learning curve by mimicking the component design of a Traditional setup (as seen in chapter 2). The manufacturer goes even further, stating: "Stanton's StanTouch®[33] technology allows you to use traditional DJ-style motions and gestures on a touch-pad style surface".

Although it is arguable that they really have mapped traditional DJ gestures with this technology, they have, nevertheless, taken some steps forward using touch metaphors to their advantage with regard to Traditional setups, as described in what they call "Slider Mode". With the "Slider Mode" touch-logic, the controllers allow the user to perform tasks that are not possible with a real potentiometer/fader, e.g., clicking directly on a slider's mark will make the value bump to that position - a situation not possible in the "real world" because a fader has to be manually dragged to the new position. Also, by holding one finger on the bottom of the slider and tapping on the top with another finger will make the slider generate the higher value for as long as that finger remains on the surface. When the second (upper) finger is removed, the slider will generate the value indicated by the first finger position.

[32] SCS.3d is also designated *DaScratch* since it is obviously targeted at scratching [3].
[33] StanTouch is the commercial designation for Stanton's two-finger multitouch technology fitted in both controllers.

Both SCS.3d and 3m enjoy a sizeable market share, because they can easily be integrated into many Hybrid solutions[34] and they also can be used to control several Virtual DJ Applications via MIDI[35].

3.6.4 Multitrack scratch controller

Fukushi has proposed a multitouch-enabled device directed at scratching tasks, depicted in Figure 3.27, one where the DJ can scratch several sources simultaneously [91], thus eliminating the time lost when scratching users switch between various turntables. Fukushi's Multitrack scratch controller is also highly effective in reducing space and components count. Much like the Attigo, the Multitrack scratch controller is based on a moving waveform metaphor, the interaction being performed via direct manipulation of the waveform, back and forth. Fukushi also proposes a new metaphor, one that enables DJs to perform record-crossfader combinations with just one finger which increases scratching performance, but also generates a new "faderless" lexicon that was not easy to some DJs; so, the learning curve, as evaluations denote, suffers.

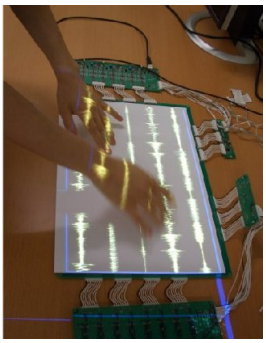


Figure 3.27: Multitouch scratch interface. [91]

3.6.5 Discussion on Multitouch Controllers

Multitouch Controllers do offer new possibilities as Virtual DJing applications' controllers, thus they stand out when compared to traditional mouse-based operations or button-based MIDI controllers[36]. They allow improvements in two directions: increasing task performance by providing both bimanual interaction [92] and with new interaction features (such as the aforementioned "slider jump"); and lowering learning curve by maintaining coherence with traditional DJ gestures, as seen on Stanton's controllers.

On the other side, touch-based controllers will always lack the tangible side that only Traditional

[34]The SCS.3d system is directly supported by: djDecks, Virtual DJ, PCDJ and Mixxx.
[35]Traktor DJ Studio, Traktor Scratch, Ableton Live, Serato Scratch
[36]Typically, MIDI controllers were based on button/triggers or sliders, rather than touch surfaces.

and Hybrid setups can offer. Also, one must keep in mind that these controllers are application-driven, thus they do not offer interaction mechanisms for realtime DJ-specific tasks, such as adding new tracks, reorganizing setup (Lemur allows it in offline operation), altering connections between DJ components (audio re-routing), and so forth - all those tasks have to be carried out in the Virtual application, using a mouse/keyboard input device.

With regard to visual feedback, these proposals have their own idiosyncrasies: Lemur and SCS are slave controllers that send messages to a master application, and therefore their visualization capabilities are often limited, and no real-feedback can be easily given to the user in realtime. On the other hand, Attigo and Multitrack-scratcher do display the song's waveform and, while it enhances task performance for scratch DJs, it degrades usability towards Club-DJs (see details in Appendix B, where a DJ Taxonomy is explained) because fine tuning and record speed syncing based upon direct hand manipulation works better with a traditional "revolving platter" metaphor.

All these proposals validate our research into architectures with a minimum of moving parts, as these ultimately ensure durability from gig to gig[37].

3.7 Setup Comparison and Conclusions

To wrap up this section, we compare all those applications and setup-types we have covered. Table 3.1 correlates the systems with a selected set of requirements: the first two requirements express the advantages of the digital over the analogue domain, namely **Signal Processing**, meaning that digital enables extensive audio effects and plugins, and **Data Storage** which ultimately allows DJs to travel with gigabytes of audio instead of kilograms of records.

In the table, **Traditional Gestures** refers to the lexicon found in the Traditional setup; having it (checked in) is of crucial importance to guarantee a low learning-curve, a requisite of our proposal. **Scratch** is self-explanatory, and is also a requisite of our proposal. **Connectivity** expresses how easy it is to extend the system, e.g., Virtual systems can be easily interconnected to new components, while Hybrids and Traditionals cannot. **Rearrangable** denotes the property that some interfaces (or real components) have which allows them to be spatially re-organized by the DJ to reach a higher level of customization. The last two items cover the transportation issues as raised by DJs with regard to Traditional and Hybrid setups: **Portability** expressing how easy a system is to transport and **Compact** denotes systems that have minimal moving parts.

If we look closely at both Virtual and Multitouch controllers, we can see that a merge between these two accommodates the typical DJ gestures and still offers the flexibility of a Virtual system. Since they

[37] This is actually a statement that Stanton uses to promote SCS sales, thus we acknowledge it as an important feature to modern DJs.

Table 3.1: Comparison of selected DJ-approaches.

	DJ Requirements							
	Digital Domain		DJing Features				Transportation Issues	
	Signal Processing	Data Storage	Traditional Gestures	Scratch	Connectivity	Rearrangeable	Compact	Portable
Traditional	✓/×[1]	✓/×[2]	✓	✓	✓/×[3]	✓/×[3]	×	×
Virtual	✓	✓	×	×[4]	✓	✓	✓	✓
Hybrid	✓	✓	✓	✓	✓/×[3]	✓/×[3]	×	×
Tangible	✓	✓	×	✓/×[5]	✓	✓	✓/×[6]	✓[7]
Wearable	✓	✓	✓/×[8]	✓	✓	×	✓	✓
Haptic	✓	✓	✓	✓	✓	✓/×[3]	×	×
Multitouch	N/A[9]	N/A[9]	✓	✓	✓	✓[10]	✓	✓[7]

[1] Only if coupled with effect-processor units.
[2] Digital Storage on Traditional systems depends on hard-disk CD players.
[3] Only to a certain extent, because it is limited by hardware.
[4] Some can perform scratch, preferably if paired with a suitable MIDI controller.
[5] Only Reactable embedded with Skipproof can perform scratch.
[6] Although these solutions are compact, they still have moving parts (the tangible objects).
[7] These run on large prototypes, but could be embedded into smaller (in height) tabletops.
[8] Not all "air-gestures" are natural, or do provide enough sensory feedback.
[9] Not Applicable because these are just controllers that target a DJ application.
[10] Multitouch can be rearrangeable, as proven by Lemur's interface.

are complementary in features, users can enjoy the benefits of unlimited tracks and digital processing, a modular and customizable system, natural interaction to DJing, digital storage medium and, finally, a compact all-in-one device solution, thus reinforcing our belief in a multitouch DJing application, as shown on Table 3.2.

Table 3.2: Merging the Virtual and Multitouch DJ-approaches.

	Signal Processing	Data Storage	Traditional Gestures	Scratch	Connectivity	Rearrangeable	Compact	Portable
Virtual	✓	✓	×	×[4]	✓	✓	✓	✓
Multitouch	N/A[9]	N/A[9]	✓	✓	✓	✓[10]	✓	✓[7]

From this perspective one may conclude that a multitouch controller paired with a Virtual system can address the issues that DJs face today, and our goal is now to propose such a system. Therefore, in the next chapter we propose an architectural layout that is able to comply with the requirements we have identified, so far.

4 MtDjing

Today's DJ uses records as building blocks, stringing them together in an improvised narrative to create a "set" - a performance - of his own.

Bill Brewster and Frank Broughton *in Last Night a DJ Saved My Life [2]*

In this chapter, we discuss challenges raised by DJ applications and propose solutions both at the architectural and interface level. At the architectural level, we propose a modular architecture that addresses the technical requirements of DJing, which we present in the first sub-section, below. At the interface level, we focus both at the interface itself (and the design challenges we had to overcome), and on user-interaction metaphors. A group of DJs has closely followed our work, and from the set of solutions that we have found, discussed, analysed and refined, our proposal was born.

4.1 DJ Technical Requirements

Our architecture was conceptualized to fulfil the set of technical requirements that must be addressed by a typical DJ application. These requirements were drawn from user expectations, analysis of related work and dictated by the DJ industry (as reported on sections 2.4 and 3.7). Building on that set of knowledge, we define the following technical requirements:

Expected Latency. Latency is widely criticised by the DJ community as one of the worst disadvantages of software vs. hardware [25], degrading both Hybrid and Virtual setups. Manufacturers aim for a low latency metric, ideally below 10ms [93; 75]. This value was set forth by Beamish et al., and is based on a study by Levitin et al. [94] that focus on measuring acceptable latency in interactive systems. The Steinberg company, which holds ASIO driver protocol[1], argues that latencies of 11 or 12 ms are acceptable[2]. It is important to guarantee this metric because it influences user experience directly and can increase user error rates when the application is used.

[1] `http://www.steinberg.net/en/company/steinberg_technology.html` accessed on 12/12/2009

[2] JDMars, "Better Latent Than Never: A long overdue discussion of audio latency issues" online at `http://www.digitalprosound.com/Htm/Articles/April/Audio_Latency.htm`, accessed on 12/12/2009.

External control. This is a common trend identified by our survey on DJing applications, it refers to the possibility of being externally controlled or, conversely, control external "devices". This duality enhances the DJ's performance by allowing control over multiple hardware and/or software components without leaving the application environment. For our application, we allow control over both MIDI and OSC protocols.

Modular hardware. DJs often interconnect gear from different manufacturers, using standard audio/-data cables. Our system also take these guidelines into consideration, providing a modular architecture that is independent from the hardware used for touch capture, audio render, external devices or visualization.

Modular software. Commercial Virtual setups popularized a couple of widely used "de facto" standards [95], such as MIDI protocol for control and the VST[3] technology for plug-in portability. The new rising standard for interconnectivity is the OSC, which is set out to replace and extend MIDI and addresses far more than just audio applications. In our proposal we strive to guarantee that all software modules/layers use standard protocols (such as OSC) for communications, thus making them easily replaceable.

4.2 Mt-Djing Architecture

In order to fulfil the aforementioned requirements, a modular architecture, depicted in Figure 4.1, was developed with all its modules separated by functionality and structured in four layers: **core** (application), **managing**, **API** and **device**. External components (both hardware and software) interact with the system through their respective APIs, while communication events are routed through the core application, thus satisfying the requirements of software/hardware decoupling and abstraction of communication protocols. The architecture is composed of several independent modules, which are briefly presented below.

Interaction Manager Receives and interprets the input from the Interaction API (a black-box component that reports touch events on the surface). This module is responsible for gesture recognition and communicating triggered events to the core application.

Visualization Manager Renders the visual feedback on the surface using the Visualization API; it is responsible for all interface management tasks.

[3]VST is the "de facto" standard plugin architecture in audio community [96]. It is owned by the Steinberg company and is a closed source product with an open development API.

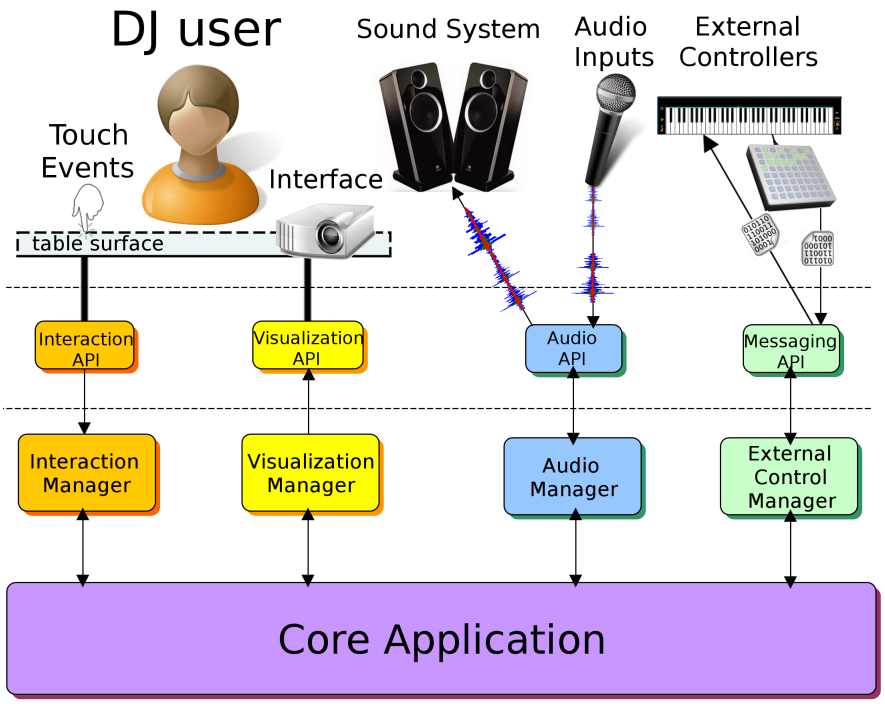


Figure 4.1: Diagram of implemented architecture.

Audio Manager Controls and manipulates audio files, using a target Audio API, and ensuring audible feedback. Receives parameters from the Core Application to modify audio in real time, and routes it through the application if an input is connected (such as microphones or other sound sources)

External Control Manager Receives and sends specific control messages (defined by the Messaging API, i.e., MIDI or OSC) to allow the core application to be externally controlled or, conversely, to control different hardware/software external instruments.

Core Application This is the main layer defined by the system, and is responsible for maintaining communications with all modules and assuring any needs for data processing. It also features an XML configurations manager, enabling DJs to easily save and share their configurations and the state of the interface (refer to Appendix A for details about our XML format).

There are two key features on our proposal, namely design **modularity** and **extensibility**. Modularity allows us to take a component offline and replace it with an equivalent software module, something that can be done because components are architected as standalone black-box modules that communicate via standardized messages and APIs. Extensibility allows the DJ application to fit alongside other hardware/software controllers that the user intends to use in his live performance, and results from adherence to the audio community standards: OSC [97; 88; 95] and the old established MIDI [42; 98; 97; 99]. As an example, one could remove the Audio Manager module and still be able to communicate with

a typical DJ software application, such as Ableton Live.

4.2.1 Module Implementation

The Core Application module is implemented in C++, the Audio Manager and External Control Manager are implemented in Pure Data [100] and the Visualization Manager is implemented in Actionscript. Maintaining a multiplatform solution was considered a requirement, since it givers users more freedom of choice, as observed in recent products such as Renoise or Mixxx.

For the **Audio Manager**, Pure Data simplified integration with VST and LADSPA[4] audio plug-ins, which can be effortlessly integrated within our application. As for the **External Control Manager**, Pure Data was chosen because it provides full access to the MIDI messaging API, allowing us to send and receive messages to/from external devices.

Both modules are designed with efficiency guidelines in mind, ensuring low latency as far as possible. There is a wide range of tools and paradigms to avoid high latency in audio applications; many developers implement their products in fully native languages which offer better performance [41](e.g. C++), while others work on parallel computing for audio [101] (e.g.: with OpenMP or using GPUs). There's also a trend, which we followed, to use RT (realtime) operating systems (such as the Linux RT kernel), as proposed in D'Groove [75] and validated by Wright et al. [102] and MacMillan et al. [103] that push Linux and Mac OS to the lead for realtime audio processing.

Separability of modules can ultimately allow the audio module is executed in a high performance environment, hence minimizing perceptible latency. To achieve the lowest latency possible we executed the Audio Manager in a real-time preemptive Linux kernel paired with an off-the-shelf USB-soundcard. Audio connections between devices and the Audio Manager are supported via the Jack protocol [104] a configuration which delivered an audio latency of $\approx$11.6 ms, measured with the Audio Manager playing several tracks simultaneously (medium load), which falls well under industry standards. For an overall latency figure, one must add the multitouch tracker's latency which, on our prototype with an optical tabletop running CCV[5] is $\approx$12ms. The remaining latency is introduced by the network, which included an off-the-shelf LAN-switch between the nodes, which introduced $\ll$1ms. Thus the full latency is $\approx$23.6ms which, as shown in chapter 5 was not perceived as a problem by the end users.

On the **Visualization Manager**, ActionScript allows us to create a full vector-based interface with a fast prototyping curve, and simplifies future design to be added later by dedicated professionals. Additionally we used Box2D[6] in conjunction with the Visualization Manager to provide realistic physics to

[4]LADSPA is a Linux opensource solution similar to VST.
[5]`http://ccv.nuigroup.com/`, accessed on 05/05/2010.
[6]`http://sourceforge.net/projects/box2dflash/`, accessed on 12/03/2010.

(a) Sheet 1 (b) Sheet 2

Figure 4.2: Low-fidelity prototype task for fader movement.

motor-torque and applied forces over the virtual turntable. The detailed class diagram of the architecture can be seen in Appendix C.

The implementation allowed us to support the desired interface and interactions that DJs needed; the multitouch metaphors we have used will be the subject of the next sub-section.

4.3 Multi-touch DJ Interactions

In this section we take a closer look at metaphors we propose to model DJ's interactions with the tabletop surface. We start by presenting a study, conducted with a low-fidelity prototype, that we performed in order to understand how DJs in our accompanying group performed certain gestures. Then, the interface concepts (inspired by the results obtained in the Chapter 3 and aiming at low-learning periods) are explained. We follow with a more in-depth look at DJ objects' behaviour, where we study issues raised when mapping certain actions into a multitouchable environment. We conclude this section introducing a toy problem that we have devised to help us in reaching a good solution for DJing.

4.3.1 Understanding DJ gestures

In order to design suitable interactions for DJs we set out to study the typical DJ gestures. As referred in the DJ performance analysis (Section 2.3) this study was based on several works such as [12; 13; 24; 8; 14; 3; 25; 26; 27; 2]. Furthermore, we used a paper prototype mock-up[7] and our accompanying group of DJ experts to test those actions.

Each DJ action was considered a task for the user. For each action, the user received two paper

[7]This methodology was inspired by the talk of Surface team-leader August de los Reyes, at Remix 09 conference.

Mixer Gestures	Fader movement Knob movement Crossfader techniques
Turntable Gestures	Place record Remove Record Rotate Turntable Move Turntable Start Turntable Stop Turntable Scratch Record Slow down Record without pitch
Routing Gestures	Create routing cable Alter routing cable Destroy routing cable

Figure 4.3: Gestures identified by the paper prototypes.

sheets with simple drawings, one was the initial state of a DJ component (e.g.: fader) while the latter was that component's state at the end of the task. Figure 4.3.1 shows the set of sheets for the task "fader move"; here the user was asked to move the first fader to an upper value. Users were asked to describe a gesture - single or multitouch, including bimanual operation - to transform sheet 1 (Figure 4.2(a)) into sheet 2 (Figure 4.2(b)). Although these gestures are very simple, altogether they form the basis of modern DJing.

From these tests we understood that all gestures (except routing) exhibited in Table 4.3 are direct manipulations of hardware components, therefore suited to direct mapping on a multitouch environment. We then created a toy problem to study adequate routing gestures, that we will address in Section 4.3.4. In the next sections we will show how this gestural lexicon is supported by the proposed interface concepts and object behaviour.

4.3.2 Outline of Interface Objects

Our UI design is based on the core concepts in the user's mental model: sound sources, records, audio manipulators (volume faders, equalizer knobs, crossfaders, and so forth) and, finally, on relationships between these objects. These concepts are directly mapped into visual representations (of the objects) that the DJ can manipulate within a live performance. With input from the accompanying group we studied how the interface can play two roles: first, maintaining consistency with the current DJ task by ensuring correlation with existing DJ gear and with identified DJ gestures; and secondly, enabling us to improve DJ tasks that benefit from virtualising some user tasks.

In Table 4.1 we present the interface objects, graphical widget (mock-up version) and list of supported features.

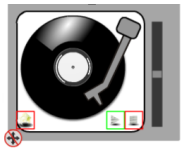
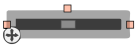
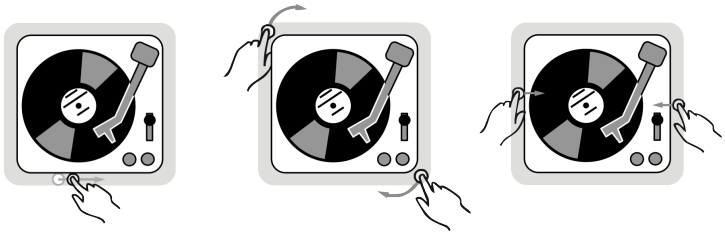
Sound Source ***(virtual turntable)***	 This object outputs the sound of the record being played. It allows direct scratch manipulation with realistic physical behaviour on the record surface. It provides a pitch slider to adjust turntable speed and several triggers to reverse direction, start/pause and stop.
Fader	 This object changes the volume of the sound being routed through it. It supports multiple touch-points, hence can accommodate novel fader gestures.
Crossfader	 This object balances the volume of two sound inputs being routed through it. It supports multiple touch-points, hence can accommodate novel fader gestures and recall basic crossfader techniques.
Triple EQ	 This equalizes the input signal with three different bands (low, mid, high) as found in typical DJ systems. It supports multitasking for each knob, thus all can be manipulated concurrently.
Record	 The record holds the sound, can be placed within the turntable platter to change the current song.
Drum Pad	 A multitouch drumpad, supporting any desired number of pads (configured with a XML file as detailed in Appendix A). Triggers play samples to emulate a typical drumpad. It was included as a request from DJs and is not usually found in DJ systems.

Table 4.1: List and description of interface objects.

Figure 4.4: The manipulation frame is represented by a grey area around the object.



(a) Move (b) Rotate (c) Scale

Figure 4.5: Three gestures for object manipulation.

4.3.3 Generic Object Behaviour

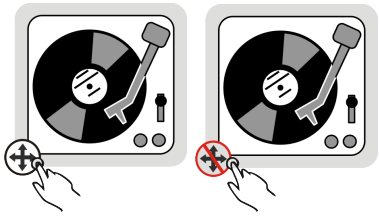
Every object inherits a set of generic behaviour (see Appendix C for a detailed explanation of Object Inheritance), allowing them to be rotated, scaled or placed anywhere in the surface canvas; this allows users to customize their setup for a DJ live session. However, that raises an issue in multitouch interactive interfaces, because a wide set of actions is available for a single input (touch action), thus it needs some sort of disambiguation.

To sort this out, we implemented a frame around the object that allows actions to be triggered with two finger gestures. Best results were observed with a single frame that supports gestures, as shown on Figure 4.4, instead of having separate handles around the object one for each action. This decision was validated with the DJ group and did not raise issues in object handling throughout the tests. We got results that are in line with Nacenta's paper [105].

Touching the frame around the object enables the user to **drag**, **rotate** or **scale**; touching the object (itself, not its frame) will trigger the appropriate DJ-actions. This solution prioritizes actions above object placement/handling, something that is quite important in a DJ performance, as confirmed by the DJ group.

Dragging the frame around with just one finger will move the (selected) object, allowing it to be placed anywhere within the canvas, as shown in Figure 4.5(a).

Figure 4.6: Combining Rotate and Scale gestures.



(a) Locking (b) Unlocking

Figure 4.7: Unlocked and locked object states.

Rotation, depicted in Figure 4.5(b), is executed with two or more fingers and is snapped to angles which are multiples of 15 degrees, as requested by users (recorded in test results).

Scaling is performed with a two finger combination on the frame, as shown in Figure 4.5(c). Users requested that a minimum/maximum size was fixed for each object to avoid error in wide/accidental motions.

A single gesture can accommodate moving, scaling and rotation, as shown in Figure 4.6. Furthermore, the DJ panel requested a **lock** feature to fix objects in place, as depicted in Figure 4.7.

4.3.4 Object Connectivity

Ensuring that the creation and performance phases do not need to be separated raises some issues in multitouch, because new objects added to the stage have to be connected to existing ones and DJs want to exercise full control on the audio flow. Much like reacTable or audioPad, our proposal uses an audio signal flow metaphor to represent object connectivity, and so the user can compose an audio flow from the source objects (those playing sounds) up to the master output.

To tackle this issue, another toy problem was created (and tested with one of the 10-user test groups). DJs were asked to perform a sequence of three tasks: first, a gesture to connect the turntable output to a fader input; then, they had to substitute the connection that goes to the fader to send the flow to an

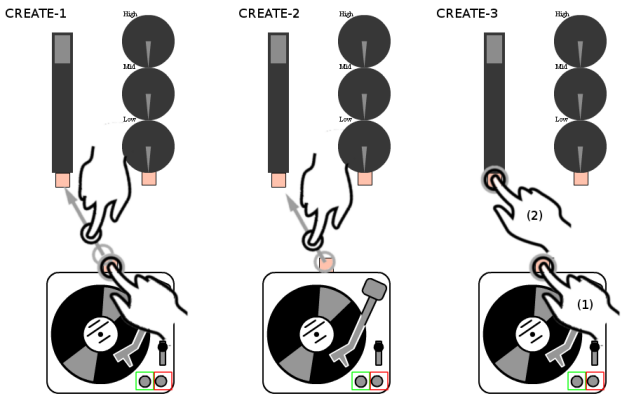


Figure 4.8: Observed gestures for creating a connection.

equalizer; and, finally, they had to destroy an active connection.

The users were asked to state what a connecting line represents; the majority (70%) interprets the line as an audio cable and would like to have that as its visual representation, with audio connectors on each endpoint. The remaining 30% thought of the line as a pointing arrow signalling the (signal) flow, and that can be easily merged with the previous representation to create a more coherent interface concept.

In Figure 4.8 we represent three different gestures performed by DJs to connect objects on the surface. Gestures CREATE-1 and CREATE-2 can be easily combined because they are mainly the same gesture, either performed with one finger, or with two fingers but leaving one still on the source object. The other gesture, CREATE-3, seems more GUI oriented, and has to be performed in two stages: the user taps on a source object and then on a target object - and a connection gets created. The CREATE-3 gesture is more error prone because the user has no feedback while the two phases are ongoing. As depicted on Figure 4.11, CREATE-2 was the most popular, followed by the CREATE-1 and CREATE-3.

The next action requires the user to alter an existing connection by switching the end point. In this task three gestures were captured, as depicted in Figure 4.9. SUBS-1 gesture is coherent with CREATE-1, it fixates a finger on the source while changing the endpoint with another touch; the line can be drawn in real-time to enhance visual feedback. The SUBS-2 gesture is also coherent with CREATE-2, as the DJ-user simply creates a new connection using exactly the same metaphor. However, the SUBS-3 gesture is not coherent (it conflicts with CREATE-2): the user drags the endpoint to the new desired location, which is exactly what CREATE-2 does to create new connections.

For the delete/destroy action three different gestures were observed, as shown on Figure 4.10. Although DEL-1 seemed a natural option for a few DJs, as shown on Figure 4.11, it is very difficult to avoid errors with this "cutting the cable" metaphor if more than one cable is in the interaction area. The

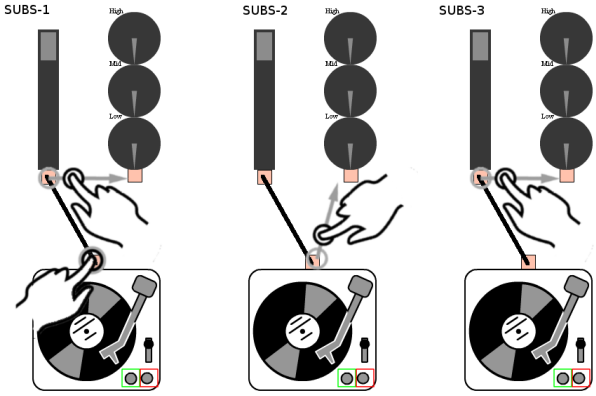


Figure 4.9: Observed gestures for altering an existing connection.

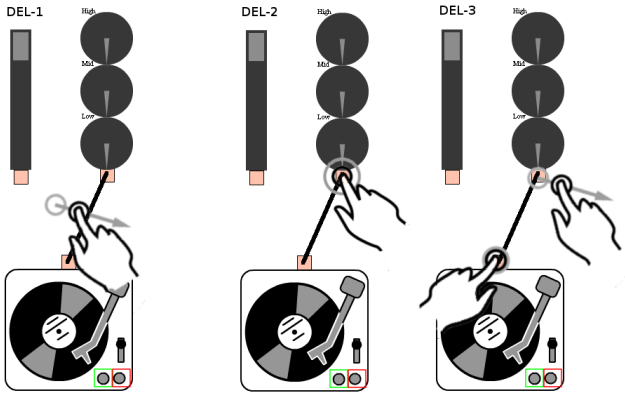


Figure 4.10: Observed gestures for deleting a connection.

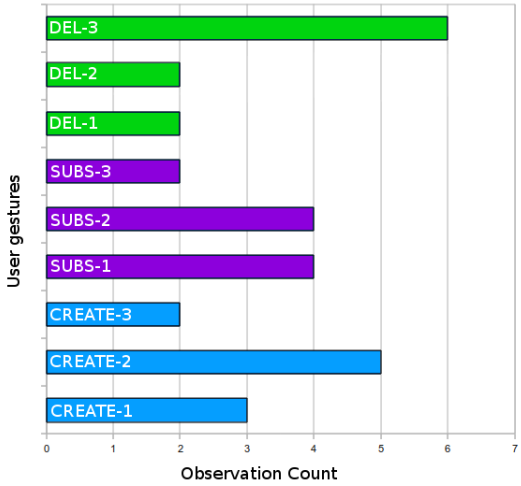


Figure 4.11: Observed gesture distribution for the toy problem.

solution for DEL-1 would be to add a selection phase previous to this gesture, so the correct cable could be highlighted before deletion. The DEL-2 gesture (a double click/tap metaphor for deleting a current connection) is very much GUI-based, and was performed by two users; however, it raises the error rate because it's easy to get a false positive or accidentally double tap an object. Finally, the DEL-3 gesture seems more appropriate for deletion, as it is coherent with CREATE-1 and SUBS-1; in DEL-3 the user fixates the source while moving the endpoint to the empty canvas, something a user does intentionally to destroy a connection.

In summary, we conclude that one finger gestures allow users to execute other tasks concurrently, using bimanual interaction, because they do not interfere with two-finger gestures. Thus, both can be combined to increase the scope of user-familiar gestures. Also, as all interface objects can be moved around, arm-reach issues raised by larger surfaces can easily be avoided.

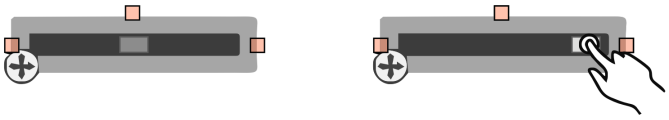
4.3.5 DJ Object Handling

Apart from generic behaviour, each DJ object has its own specific gestures. In the present section we describe the behaviour of the **fader**, **EQ**, and **turntable**, highlighting features that improve DJ performance. This lexicon is drawn directly from the low-fidelity prototype presented in Section 4.3.1.

In **mixer gestures** all DJs were very coherent, all gestures aiming direct manipulation of fader/knob elements; the only idiosyncrasy being how DJs interpreted the "grabbing" concept. Figure 4.12(a) shows how some DJ-users simply touch the components because they could map that concept directly into a touchable surface, while Figure 4.12(b) highlights another modality were the DJ grabs the fader as if it was physical. The behaviour was then added to the interface logic, but later on we discovered that DJs

(a) One finger (b) Fader grab

Figure 4.12: Two different fader interactions.



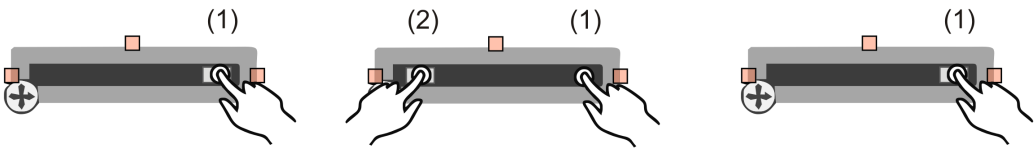
(a) Before touch (b) Instant jump

Figure 4.13: Fader jumps instantly when touched.

did not exhibit it in multitouch environments, and so it was discarded.

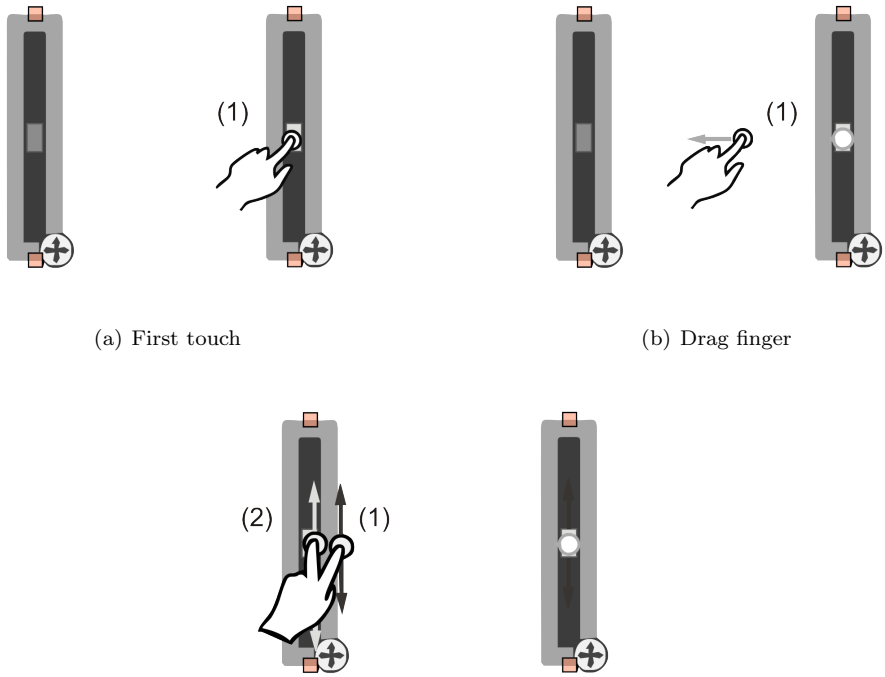
We also improved DJ faders by supporting new features such as instant-jump, multiple touch points and hold-down control. Instant jump, for example, is a feature that allows users to touch any point within the fader and have it instantly jump to the new position without having to drag the fader cap manually; it enables fast crossovers, as depicted in Figure 4.13(a), and is also found in our **EQ** objects, allowing us to raise/drop the equalization faster than with the Traditional knobs - a feature denoted "EQ kill" which is only available in certain mixers via a trigger button.

Multiple-finger support on fader objects allows users to jump from one point to another and, later on, easily recall the first position. This behaviour is fully scalable to how many fingers are placed within the



(a) Single Touch (b) Second Touch (c) Recall first position

Figure 4.14: Fader stores touched points and recalls them in order.



(a) First touch

(b) Drag finger

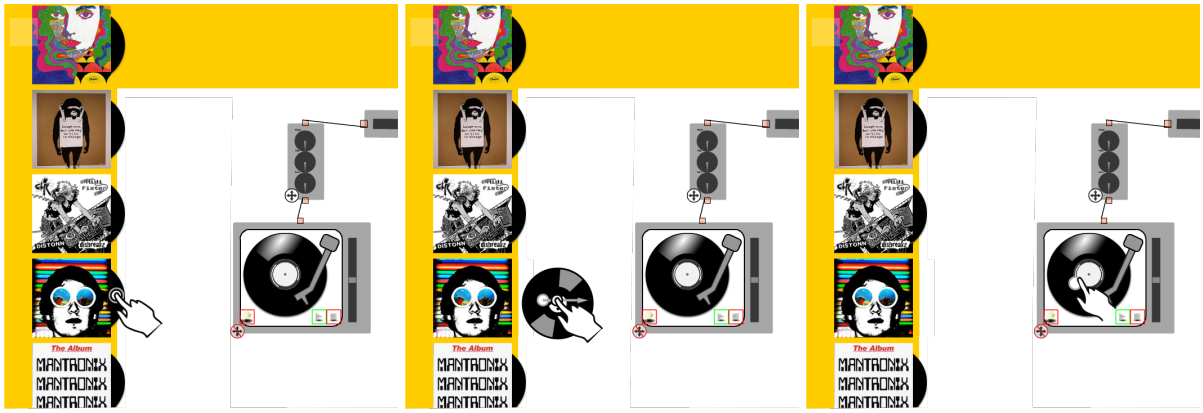
(c) Control multiple faders

Figure 4.15: Touching one or more faders allows us to control them with a single gesture.

fader, and Figure 4.14 shows an example with two points: in Figure 4.14(a) the user touches the fader for the first time; then, a second touch forces a jump to a new position, as depicted in Figure 4.14(b). Later on, the second finger is lifted and the first fader position is instantly recalled, as shown in Figure 4.14(c).

A last advantage of our fader's implementation is the ability to control several faders at once with the same gesture, as requested by the users during a test session with the first prototype. Figure 4.15 shows how to activate this feature: while touching a fader, the user can drag his finger around the canvas and still maintain control over the fader, up until that same finger is raised, as depicted in Figure 4.15(b). But if the user touches another fader with a new finger, that same behaviour is observed, thus the user can execute interesting motions, such as parallel control over two objects with the same hand, as depicted in Figure 4.15(c). Remarkably, our DJ testers took this feature even further, because if one of the two faders of Figure 4.15(c) is rotated 180 degrees, when the user moves both fingers in parallel up/down it is actually switching between those audio channels, without the need for the crossfader object.

For the **turntable gestures** different concepts were observed. Changing the record in the paper mock-up was trivial, thus mapped to direct manipulation: dragging a record onto the turntable's picture, as depicted in Figure 4.16. Removing it was not mapped in the prototype, since it is not meaningful



(a) Grab the record. (b) Drag to desired position. (c) Drop the record.

Figure 4.16: Changing a record in realtime.

(a) Traditional situation. (b) Multitouch interaction.

Figure 4.17: Applying forces to speed up the record.

in the virtual context. To increase user friendliness, if a record is dropped by mistake on a component other than a turntable, it will automatically return to its original position (the side panel). Conversely, if a record is dropped on top of the turntable widget its size will match that of the platter, conveying the idea that it was placed correctly, as depicted in Figure 4.16(b).

As for direct manipulating the audio through the record, two actions were identified in Table 4.3: scratch and slowing/accelerating speed. As observed in user sessions, scratch maps directly to moving the record back and forth with the hand/fingers. Altering speed is usually achieved by DJs holding the label of the record, causing it to slow down (a common technique in beatmatching); this behaviour is exhibited in Figure 4.18(a). Conversely, to slightly increase the record's speed (e.g., to perform a beat alignment between two different tracks), DJs push the record forward with fast motions, as shown in Figure 4.17(a). As we have implemented a physics based simulation, DJs will naturally use these metaphors and apply forces to different places in the record to obtain those effects, as depicted in Figure 4.17(b) and 4.18(b).

(a) Traditional situation.

(b) Multitouch interaction.

Figure 4.18: Applying forces to slow down the record.

4.3.6 Interface Canvas

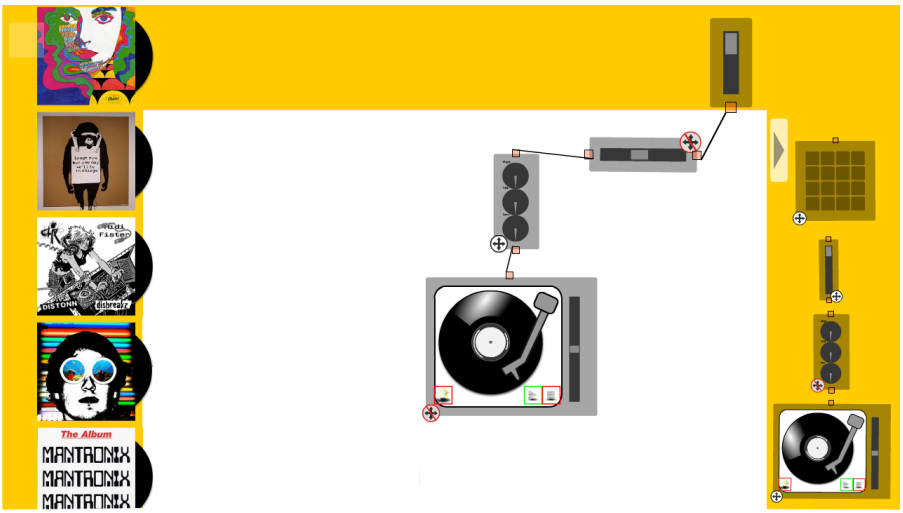


Figure 4.19: Overview of the Interface.

Figure 4.19 shows our proposal's interface; it has a canvas (white area in the center) and panels. In order to allow users to take full advantage of the interactive surface, panels can be hidden by touching the arrow handle, as depicted in Figure 4.21. Conversely, to restore an hidden panel the arrow handle will exhibit the opposite direction and, if touched, causes the panel to reappear.

The top panel allows manipulation of the master output channel. For the final version, we have included a new set of features such as (a draft of) waveform visualization, equalization, panning and a headphone output for the pre-listening channel, as depicted in Figure 4.20. This feature provides yet another advantage over typical systems, because by creating connection cables to the headphone object, the user has the freedom to listen to any point in the audio route.

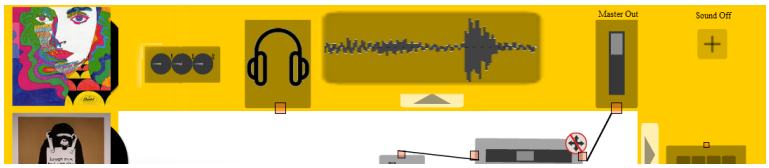


Figure 4.20: Top panel with Master Channel details.

(a) Touching the panel handler. (b) Panel hides.

Figure 4.21: Interaction for hiding side panels.

Finally, with the help of a DJ who's also an experienced designer, a new interface draft was proposed, as shown in Figure 4.22. As our architecture is fully vector-based (built on standard SVG components), supporting new designs is very straightforward, thus making the prototype easily "skinnable".

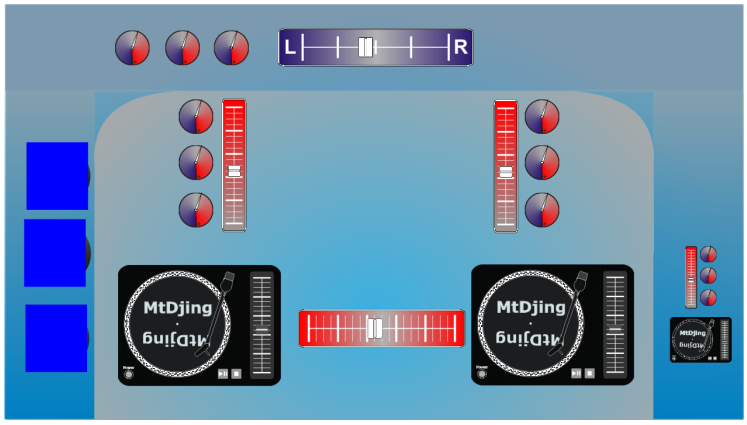


Figure 4.22: Draft of the new Interface proposal.

4.4 Extending Interactions

As described throughout Chapters 2 and 3, DJs typically add external input controllers to their performance, some using foot-controllers, drumpads or even professional software solutions. Here we show how new tools can be easily integrated within the Mt-Djing application, due to open interconnectivity and standard-compliance. The following examples were brought by some DJs in our accompanying group that actually use these controllers as a part of their performance, typically to enhance bimanual interaction to hand-foot combinations, or to perform musical tasks that require professional weighted keys or drumpads.

4.4.1 External Control

In order to provide a higher degree of control to professional DJs several footcontrollers have been produced by top manufacturer's. Here we show how these foot-interfaces can extend bimanual interaction and enhance DJ performance within our prototype; and we present other MIDI-capable devices that can fit within MT-Djing. The first exaemple shows a typical off-the-shelf MIDI footcontroller, while the second is a proposal by the same author to create a OSC/MIDI controller with gestural support. The latter example depicts two hardware devices that can enhance control, namely via pressure sensitive pads.

4.4.1.1 Footcontrollers

Figure 4.23: Using the DJ application with a MIDI footcontroller.

Figure 4.23 shows a DJ expert combining feet and bimanual interaction in our proposal. Here the DJs opted to assign MIDI messages to the foot-switches so that they could automatically load new songs

in each turntable. The foot-pedals were also mapped to the interface MIDI messages and provided control to each of the vertical faders. Ultimately this type of configurations has to come from users' preference.

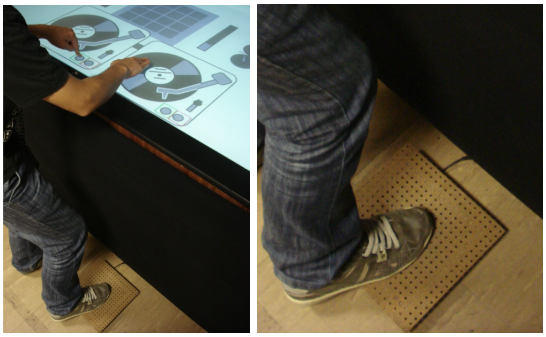


Figure 4.24: Augmented DJ application.

Another foot-controller is proposed in [106], as depicted in Figure 4.24. This conventional wooden board has been fitted with lost-cost microphones to allow gesture recognition, thus extending Mt-Djing to a multimodal situation where feet-gestures trigger DJ-specific actions. For test purposes, Figure 4.25 gestures were mapped. The first resembles the natural tapping motion of the DJs and musicians reacting to a song's *tempo*, which sets a metronome at the desired speed. The latter, allows users to turn the metronome sound on or off, by dragging their feet on the board. These techniques allowed DJs to use natural interaction to a common technique called tap-*tempo*, mostly found in Virtual DJ setup, that can adapt the songs *tempi* to a user-defined tapping beat.

Figure 4.25: (a) Tap gesture; (b) Drag gesture.

4.4.1.2 Drumpads

Throughout the sessions with the DJs, they expressed the importance of MIDI controllers to modern DJing. Thus we experimented with several devices that the DJ experts brought to the final test sessions. Here we show Mt-Djing paired with both a fader-alike controller and a professional MIDI drumpad as depicted in Figure 4.26. The connectivity is straightforward since these devices send standard MIDI messages, thus are able to control any parameter of our Audio Module.

Figure 4.26: Interacting with MIDI faders and pad.

When lacking multitouch surfaces with pressure-sensitivity, a topic not addressed within our work until this stage, the DJs agreed that the external drumpad fitted nicely into the Mt-Djing setup. Unfortunately highly-sensitive multitouch setups will ultimately detect the controllers as a "touch-point", this can create accidental input, thus we propose that such on-surface devices would only be fitted with touch-surfaces that need pressure to be activated. Another suitable possibility would be to detect these controllers via optical or RFID tracking.

4.4.2 Integration with the professional DJ workbench

The majority of commercial software does not allow to map all functionalities via MIDI, such as track selection and dynamic routing. But the remaining features they allow via MIDI present exciting possibilities for DJing purposes.

Integrating Mt-DJing with existing DJ workbenches is straightforward, and facilitated by the usage of MIDI protocol in the External Control manager. In this section we show a possibility in replacing the Audio Module with a professional solution.

In Figure 4.27 our solution is acting as a MIDI controller for Ableton Live (discussed in Section 3.1) allowing DJs to control musical parameters such as individual channel volume, playout, equalizers and master channel. Our DJ experts suggested some features that would ease integration with these existing workbenches, even including Traditional setups with MIDI-capable components, that we will present and discuss in Chapter 7.

Figure 4.27: Interacting with Ableton Live via MIDI mesages.

4.5 Summary

We presented our architecture and modules that compose the Mt-Djing, these assure that the identified DJing requirements are met.

For the remainder of the Chapter we focused on describing the metaphors we designed for supporting DJs gestures; this lexicon was studied via a low-fidelity prototype with the accompanying group and improved through test results described throughout Chapter 5. In order to convey a suitable metaphor to object connectivity issue, it was isolated in a toy problem and tested with a panel of DJs, concluding upon the best gestures to create/alter/destroy lines between objects.

We followed with a through listing of our interface concepts, description of DJ object handling (e.g.: physics-based turntable behaviour) and detailed the advantages of our setup over other products and proposals we covered. These advantages (e.g.:multi-finger support for faders and dynamic audio routing) will later translate into task and creativity enhancements. Also we describe the interface canvas and present a new possibility for the interface design.

Concluding upon the selected multitouch tabletop to host our solution, it brings a handful of advantages that can be used to maintain relationships with the user mental model and enhance musical expression level, namely:

Horizontal interaction. DJing with analogue gear has always been performed on a horizontal table. Therefore, the tabletop is a suitable surface for the DJ task.

Natural Input. Multitouch tables accept the exact input that real systems have, human-hand/finger touch, and output a visual representation of real systems.

Multitask operation. Research has demonstrated that parallelism that results from two handed-interactions can increase task performance [107; 92].

This builds up the desired assumptions that we will test in the following Chapter, meaning that our solution can fit amongst DJ expectations of those acquainted with Traditional and Hybrid setups, and improve performance when compared to Virtual setups. We also foresee that because multitouch surfaces lack haptic feedback our solution will yield worst results for Scratch experts.

5 Evaluation

5.1 Test Environment

Tests were structured in three stages: a pre-test questionnaire to determine the DJ's profile and experience regarding multitouch devices; three DJ-oriented tasks; and, finally, a post-test questionnaire to rate the prototype's ease of usage and an interview to get detailed information about interaction issues. With the users' permission tests were videotaped, application audio was recorded, interviews were transcribed from audio recording and the application logged events (with timestamps) to a text file, for detailed error analysis.

5.1.1 Participants

Evaluation was carried out using a panel of ten DJs, four of them amateurs with two years of experience and six semi/professional DJs, with up to twenty years of knowledge. These DJs were not part of the aforementioned accompanying group to ensure that no previous knowledge would interfere with the test outcome. From our survey on DJ performance, we understood that different styles of DJing have specific application-requirements, and result in different DJ performances; therefore we included Scratch and Club/Radio-DJs in the group, as depicted in Figure 5.1.

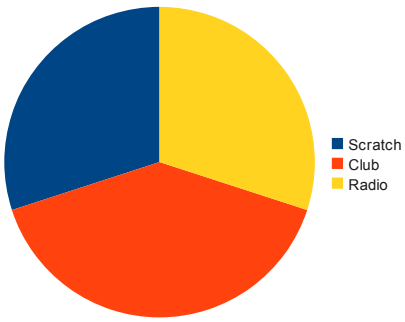


Figure 5.1: Distribution of DJ-styles for panel of DJ-testers.

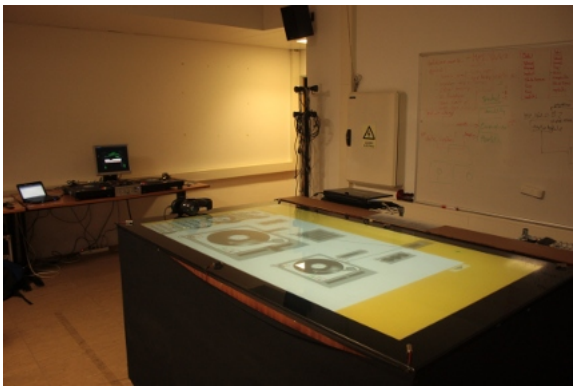


Figure 5.2: Multimedia room prepared for final user testing.

5.1.2 Apparatus

All tests were conducted in a closed environment. Sala João Lourenço Fernandes[1] on the IST Tagus Park is a large multimedia room with a 5.1 surround system, an A0 optical tabletop (together with its accompanying computer), a dedicated soundcard and DJ headphones. A video-camera recorded the test sessions, while a dedicated hardisk-recorder (with a pair of condenser microphones) captured audible output and user comments for later analysis.

On the second round of tests DJs were allowed to discuss and compare the Mt-Djing prototype to Traditional, Hybrid and Virtual systems that were brought into the room. Figure 5.2 illustrates these setups, with Virtual (Mixxx), Traditional (two turntables and a mixer) and Hybrid (Mixxx with vinyl tracking). A more detailed description of the technical setup is available in Appendix B.

5.2 Preliminary Evaluation

In the first set of tests we used an earlier version of the Mt-Djing prototype which lacked multitouch faders, a browsable catalogue, the possibility of adding new objects in realtime, and dynamic connectivity. However, the prototype had the relevant core features needed to study how ease was for DJs to use it, when compared with their expectations and with the three major setups: Traditional, Hybrid and Virtual.

5.2.1 Test Description

The tasks were designed to evaluate the basics of modern DJing, thus essentially focussed on mixing songs. For an homogeneous analysis, all DJs used the same two songs on each task (mid tempo songs,

[1] `http://lab-jlf.ist.utl.pt` accessed on 7/10/2010.

around 100 BPM). The first task (**T1**) was designed to understand how DJs operate a progressive mixing, by aligning two songs together, with no restrictions given at this stage. Thus, on this task, the DJ was free to use all the interface objects available on the test-prototype, namely: channel faders, cross-faders, standard DJ triple equalizers and sound sources, visually represented by two turntables. This freedom of choice allowed us to understand which objects are preferred for a typical mix, with regard to different DJ styles. The second task (**T2**) studies a restricted form of interaction - we asked DJs to mix the two songs using only the faders, without equalization. And finally, for the third task (**T3**), we set out to study fast DJ gestures, such as fast cross-fading between two songs and consecutively stopping and starting two different songs; this test allows us to rate the application's adequacy for a more scratch/turntablist performance.

DJs were asked to rate each task's perceived ease of use when the prototype is compared to systems they regularly use (Traditional, Virtual and Hybrid). The scale was set from 1 to 5, with the lowest score implying that the prototype is much harder to use, while the maximum score shows the opposite, i.e., our prototype being much easier to use. Multi-tasking was also accounted for, through video analysis and data logging, to locate actions where the user took advantage of bimanual interaction to simultaneously perform different actions.

5.2.2 Results and Discussion

Evaluation allowed us to gather some interesting results. In Figure 5.3 we present a comparison of the scores given by the panel of DJs experts after completing the tests. We noticed that, with the given tasks, there was no meaningful difference in the ratings against the Traditional or Hybrid, so both were collapsed into a single variable. However if tests were based on a different set of tasks, such as record selection, Traditional and Hybrid setups should yield different results when compared to our system.

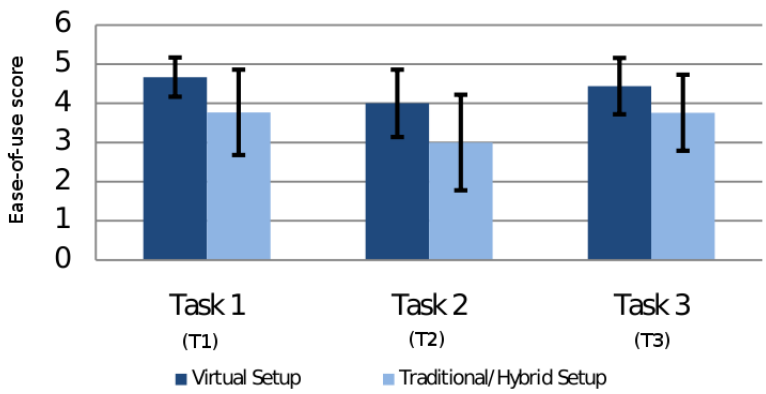


Figure 5.3: Comparison of task's ease-of-use: our system vs. typical DJ setups.

Figure 5.3 denotes mean values for each score showing that, overall, our system is appropriate for any DJ who is familiar with any of the three setups, because results lie above the medium score - which means that the system is at least as easy to use as the one its been compared to. The standard deviation (σ) give us a clear insight of the fluctuations of the scores for each task.

Comparisons with the Virtual setup turned out to be very good: fluctuations on the test scores are low, while mean value is always above 4 points, thus ensuring that DJs found advantages in our proposal when compared to the Virtual setup. Against the Traditional (or Hybrid) setup, the σ shows a more meaningful value, making the scores fluctuate higher. Analysing each DJ's score set in comparison to the Traditional, the lowest scores were rated by DJ Scratch experts, while Club DJs found the system in tune with their expectations. This confirms the hypothesis that a multitouch device will suit a wide number of DJ styles (such as Club or Radio) but will fail against the mental model of scratch DJs.

Also, when looking in detail at **T1** and **T3** for Traditional/Hybrid setups, we see that σ has decreased in **T3** because the majority of the DJs found that it was faster to manipulate a multitouchable fader than an analogue one[2], when it comes for simple crossfading between songs. The highest rate of bimanual interactions was detected when DJs needed smooth transitions, specially using fader-knob combinations on each hand, which relates directly with **T1**. The low score in **T2** allows us to conclude that we need to enhance our fader's capabilities, which we have done on the next prototype version, as reported in Section 4.3.5 (e.g.: multi-finger support for the faders).

On a final note, we believe these results show that the interface was built using DJ core concepts, and that it can easily be used by DJ experts. Remarkably, although 60% of users had no multitouch experience whatsoever, and the remaining percentile uses multitouch mobile-phones or drawing tablets, we found no statistical correlation between the scores of these two groups. Also, users successfully accomplished all tasks within their own time/quality expectations.

5.3 Final Evaluation

Latter evaluation with a panel of DJ experts allowed a better pitch of our solution against all major setups. We used an improved prototype, whose features are described in Chapter 4. Turntable behaviour was improved with a physics-based simulation to accommodate the typical torque metaphor that DJs are acquainted with.

[2] As mentioned in the Interaction section, the majority of DJs understood that they can trigger fast motions by touching the desired endpoint for the fader position, instead of having to drag it from origin to endpoint.

5.3.1 Test Description

For the final evaluation, the tests focused on mixing and beatmatching pairs of songs selected for their overall similarity, although with different tempi (ranging from 100-120 BPM). These songs were previously tested by a couple of DJs from the accompanying group to ensure that they had the same technical difficulty level and were indeed matchable. Songs were randomly selected from our song pool, to guarantee that DJs could not memorize the correct pitch values to speed up the alignment task, in any test.

Test-DJs were informed that no aesthetics judgement on the mix would be performed, as well as any skill-evaluation or score. In fact, DJs had to verbally inform us when they felt that both both songs were aligned and the mix was completed. Latter on, with video analysis, we asked two DJs of the accompanying group to help us in setting the tasks' start and end points, to get some homogeneity in the results.

The test session had six tasks; the first five aimed to mix/beatmatch a pair of songs in each setup, while the final task offered DJs a 5 minute session in our prototype, with songs of their choice. The first five tasks allowed us to develop a novel comparison between setups, and are denoted as follows:

- **V**: mixing on the Virtual setup
- **T**: mixing on the Traditional setup
- **H**: mixing on the Hybrid setup
- **Mt-p**: mixing on Mt-Djing, but aligning only with pitch sliders
- **Mt-full**: mixing on Mt-Djing, but aligning with pitch sliders and direct manipulation over the record motion

The separation of the prototype in two different tasks, one using only pitch sliders while the other also included direct record manipulation, was done in order to study how DJs perceived the implemented turntable behaviour. As we had previously identified problems raised by the lack of haptic feedback, this was considered an important aspect in our final evaluation, and thus was tested separately.

Although we also employed questionnaires at this stage (described further down in this chapter), these tests conveyed us quantitative, rather than qualitative data, as we focused (for each setup) on the elapsed time needed to complete the tasks at hand.

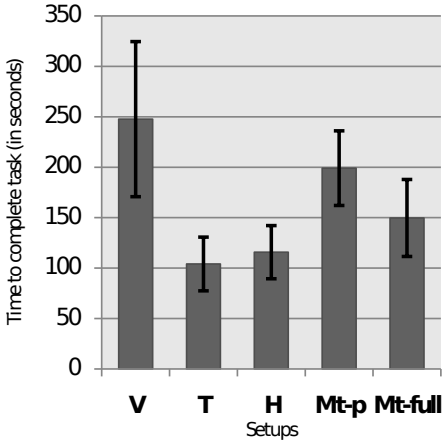


Figure 5.4: Average time needed time to complete the tasks for each setup.

5.3.2 Results and Discussion

From the test results we compute both the average and the σ of the elapsed time for each task in every setup, as depicted in Figure 5.4. Our prototypes' (**Mt-p** and **Mt-full**) results are much better than the results for Virtual setups (**V**) with over less 100 seconds of elapsed time; proving that our setup is indeed more natural than the Virtual. But, as expected, both Mt-p and Mt-full took about 30 seconds more in comparison with Traditional (**T**) and Hybrid (**H**), since the majority of our expert DJs have been using them for many years.

A detailed σ-comparison between all setups in Figure 5.4 validates not only the previous statement but also our DJ taxonomy (see Appendix B), since DJs that classified themselves in a category got similar results (elapsed time) as others in the same class. **T** and **H** show a standard deviation of $\approx$26, while **V** ranks much higher, 76.89, because scratch users are not familiarised with this setup. **Mt-p** and **Mt-full** around $\approx$38, showing that overall it is easier for all types of DJs than the Virtual. This also shows that a separation, accordingly to Appendix B, is needed in order to evaluate the solution more precisely.

The average values shown Figure 5.5 enables us to conclude that Club and Radio DJs operate quite well with our solution, showing serious improvements when compared to results in **V**. For Scratch experts, shown in Figure 5.6, we see that this class of DJs only accomplished better than **V** in **Mt-full**, meaning that they are much more efficient with direct record manipulation. Indeed in **Mt-full** they exhibited a result very close to **T** and **H**, enabling us to conclude that the physical behaviour is a valuable feature and conveys a higher natural interaction. When restricted to aligning tracks with the pitch sliders, shown on **Mt-p**, they recorded lesser results than any of the setups, showing that Scratch-DJs are not experienced with this paradigm for beatmatching.

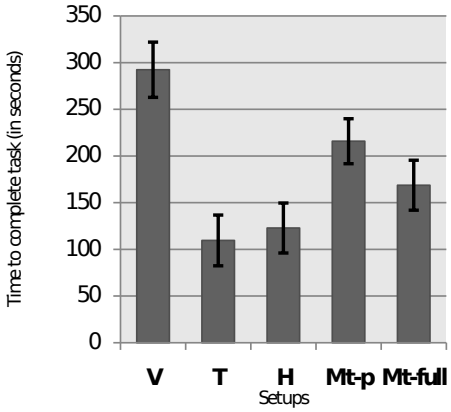


Figure 5.5: Average time needed for Club and Radio-DJs to complete the given tasks for each setup.

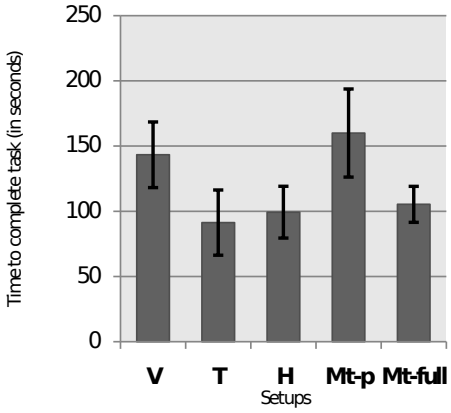


Figure 5.6: Average time needed for Scratch-DJs to complete the given tasks for each setup.

Also this separation makes sense because, as identified in Chapter 2, Scratch-DJs (Figure 5.6) tend to mix much faster than Club or Radio DJs (Figure 5.5) even with the same set of songs and instructions.

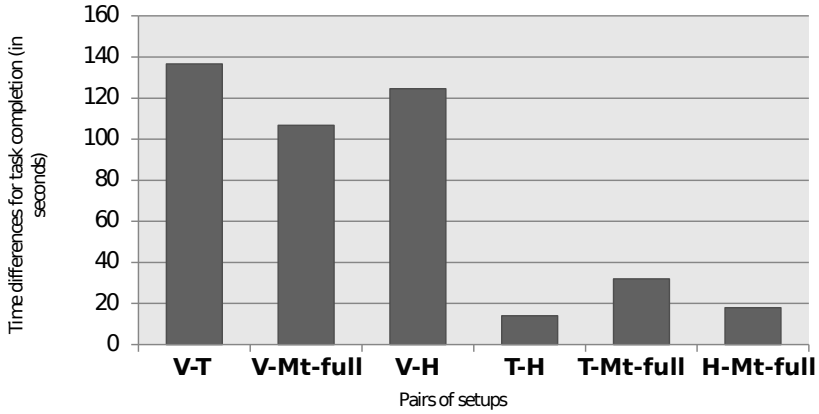


Figure 5.7: Time differences to complete tasks, between pairs of setups.

In order to draw a final conclusion on setup comparison, we must account the time differences for each user in each setup-pair, depicted in Figure 5.7. It is easy to observe that the Virtual exhibited worst results in any of the comparisons, and in **V↔Mt-full** we observe our solution providing an average of 100 seconds in task improvement. This concludes upon the adequacy of multitouch towards DJing, and shows that touch support, bimanual and horizontal interaction help users in achieving better results than with Virtual Setups.

As far as comparing our setup to the Traditional and Hybrid, we collected optimistic results. **T↔Mt-full** and **H↔Mt-full** show that DJs mixed an average of 33.9 seconds faster in the Traditional and about 45.6 seconds faster in the Hybrid. A value not surprising since we are virtualzing the assets of the Traditional/Hybrid setup and providing a less haptic interaction, but with digital advantages.

Finally **T↔H** show a slight, almost irrelevant variation, because users tend to use it solely with the traditional components, not even utilizing the computer (which is left only with the responsibility of tracking the timecoded vinyl as explained in Chapter 3).

5.4 Questionnaire and Open Speech Comments

Throughout the test sessions several interesting remarks were pinpointed by DJs; as a consequence, we identified new features that are of extreme importance to further develop our DJ solution.

The overall feeling of DJs towards our multitouch proposal was optimistic. All users were keen to stress out the advantages of both bimanual interaction and multi-finger manipulation of the fader components. However, some remarks were made on our prototype's multitouch tabletop excessive sensitivity,

prone to accidental activation; our proposal is to use technologies such as FTIR or capacitive that only react to more explicit touches.

All users, including those who had no previous multitouch experience, mentioned that the interface was very easy to use, and its concepts were simple ones. Manipulating objects around the canvas was recognised as a highly valuable feature for DJ users that want to exercise creativity in setup configuration. Also, all DJs agreed on the fact that "virtual cables" should be hidden/shown by request (a toggle switch), as users that create setups understand their setup's audio flows even when cables are hidden.

A number of other comments were noted as far as some not implemented features, namely visual feedback of the values of each component (e.g.: percentage value of the fader). This was latter corrected and included in the interface design, as previously shown in Section 4.3.6 were fader components provide a higher degree of visual feedback, much like the Traditional setup.

5.5 Summary

We conclude this section with a brief summary, highlighting our own contributions.

On the first stage we focused on testing how our proposal's ease of use fared against the Traditional, Virtual and Hybrid setups. Latter on, using an improved version of the prototype, a group of DJ experts compared our setup with all the others using the same test sets and under the same conditions.

Our results validate our hypothesis, namely that a gadget-free multitouch interaction allow us to convey natural DJ metaphors, suitable for the majority of DJ-styles. Our proposal's lack of haptic feedback is its biggest disadvantage, and excludes Scratch-DJs from the set of possible users. All others, including Club and Radio-DJs, found the system in tune with their expectations. This is corroborated by test results, which show that our solution fares better than Virtual setups, and is at pair with Traditional and Hybrids - both in ease-of-use and in the time to complete the tasks at hand.

Finally, through interviews and recorded speech, we denote how proposed enhancements such as multi-finger support for faders and realtime connections were exciting news to DJ experts that are at ease with state-of-the art commercial solutions.

6 Conclusions

For our work, a broad range of DJ-related solutions has been studied; however, we found that few of them address issues raised by the mappings of actions on the virtual setup into a touchable environment. With extensive input from an accompanying group of DJs we proposed a multitouch solution, one that bridges the gap between Virtual and Traditional setups and offers the advantages of both setups, using a single interactive surface which provides a gadget-free interaction.

Our prototype was developed taking DJ requirements in consideration, and is a modular application that can be either integrated in modern DJ workbenches, or be used as an all-in-one solution. After experimenting with paper mock-up prototypes, analysing DJ performance, studying literature, and conducting interviews with experts DJs we have designed an interface that supports natural interaction in DJing. Our proposal has several strong points, such as a fully dynamic and customizable interface, multi-finger support for faders, physics-based turntable behaviour and MIDI/OSC control, to name a few; this improves some of the traditional tasks and allow DJs more freedom to exercise their creativity.

To evaluate the adequacy of multitouch towards the DJing context, we had a panel of DJ experts running our system in two phases. In the Preliminary Evaluation phase we evaluated interaction metaphors as a whole, and studied issues related to mapping problems for touch-based interaction, such as dynamic object connectivity. In the Final Evaluation phase, we evaluated the identified DJ setups (Virtual, Traditional and Hybrid) against our proposal; we also made a novel contribution to this subject area by comparing all the aforementioned setups, something that, as far as we know, has never been done.

The results gathered allow us to conclude that our proposal suits both expectations and needs of Club and Radio-DJs, but fails against the mental model of Scratch-DJs due to the lack of haptic feedback of turntable motion. Tests show that Mt-Djing fared better than Virtual setups for all DJs, both in ease-of-use and task duration, which was reduced by an average of 100 seconds. As for tests against Traditional and Hybrid, our solution scored a similar ease-of-use but slowed DJ tasks around 30 to 40 seconds. Our proposal has been quite favourably reviewed by DJ experts, which also contributed with additional comments, and have helped us in validating our interface concepts vis-a-vis the DJ's mental models.

The fact that DJ users grasped the concepts and showed excitement over our novel task enhancements is most rewarding, and ultimately shows that our work had a positive impact, and will hopefully be a reference to researchers on these subjects.

7 Future Work

Our work essentially studies touch-based interactions within the DJ context, leaving out many other interesting paradigms such as tangible or mixed reality scenarios. Time constraints (including time needed to develop a prototype to test our assumptions) also dictated that other problems could not be taken into consideration; these include issues related with object connectivity in the DJ context, e.g., adding new objects directly onto a connection line could automatically trigger the system to insert them between existing ones, something that would clearly decrease the time needed to create new setups. Other welcome additions would be providing user feedback as to what connections are possible for each object, as available in some commercial products (e.g.: Reason), and highlighting all possible points for connection "docking" when the user is dragging a connection around the canvas.

Much can be improved in terms of visualisation, as our work was essentially focused on the delivery of a working prototype to test the basic assumptions of modern Djing. As examples of DJ requests that we did not implement, one is rendering an audio visualization directly in each connecting line to help the user to understand audio flow, in the same vein as the Reactable; another is providing a panel, underneath each object, that could be opened to access advanced features, mimicking side panel behaviour.

Although the prototype was primarily targeted at medium/large multitouch tabletops (roughly measuring about the same as two turntables and a mixer), porting it to other platforms is certainly possible; one can imagine how hand-held devices are exciting possibilities for DJ users. Again, using Reactable as an example, it was recently ported to small mobile devices[1], while maintaining the majority of concepts, and allowing the canvas to be zoomed in/out. On the other hand, our gestures were developed taking into consideration the Traditional lexicon of a DJ session, thus, for new platforms, some would have to be re-designed while others would be left out.

A wide range of different features could provide more DJ objects for the prototype, as new ideas blossom faster than we can implement them; we highlight new turntable objects, e.g., with conveyor-belt waveforms [90; 91] that probably would provide Scratch-DJs with more precise sound manipulation.

Finally, as the reader has probably guessed, many other topics were themselves dissertation-worthy and could not be taken into consideration, such as navigating through large music collections.

[1] `http://www.reactable.com/products/mobile/`, accessed on 7/10/2010

Bibliography

[1] Hegarty, P.: Noise/Music: A History. Continuum Books (2007)

[2] Brewster, B., Broughton, F.: Last Nigh a DJ Saved My Life: The history of the disc jockey. headline, New York, NY, USA (1999) Originally issue in '99 and New Centenary edition on 2006.

[3] Hansen, K.F.: The acoustics and performance of DJ scratching. Analysis and modeling. PhD thesis, Kungl Tekniska Högskolan (feb 2010)

[4] Hansen, K.F.: Turntablisme - his master's voice: The art of the record player. Master's thesis, Norwegian University of Science and Technology, Trondheim, Trondheim, Norway (1999)

[5] Bell, P.: Interrogating the Live: A DJ Perspective. PhD thesis, Newcastle University (2009)

[6] Cooper, M.: Hip Hop Files: Photographs, 1979-1984. From Here to Fame (2004)

[7] Kugelberg, J.: Born in the Bronx: A Visual Record of the Early Days of Hip Hop. Universe (2007)

[8] Hansen, K.F.: Playing the turntable: An introduction to scratching. TMH-QPSR **42**(1) (2001) 069–079

[9] Slayden, A., Spasojevic, M., Hans, M., Smith, M.: The djammer: "air-scratching" and freeing the dj to join the party. In: CHI '05: CHI '05 extended abstracts on Human factors in computing systems, New York, NY, USA, ACM (2005) 1789–1792

[10] Goldman, A.: Disco. New York. Hawthorn Books, New York (1978)

[11] Jones, A., Kantonen, J.: Saturday Night Forever: The Story of Disco. A Cappella Books, Chicago, Illinois (1999)

[12] Pray, D.: Scratch. Video Documentary (November 2001)

[13] Hansen, K.F., Bresin, R. LNCS 2915. In: Analysis of a genuine scratch performance. Selected revised papers, berlin heidelberg edn. Springer Verlag, Genova, Italy (apr 2004) 519–528

[14] Lippit, T.M.: Turntable music in the digital era: designing alternative tools for new turntable expression. In: NIME '06: Proceedings of the 2006 conference on New interfaces for musical expression, Paris, France, France, IRCAM — Centre Pompidou (2006) 71–74

[15] Beamish, T., Maclean, K., Fels, S.: Manipulating music: multimodal interaction for djs. In: CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2004) 327–334

[16] Cross, D.: A History of the Development of the DJ Mixer features; an S&TS perspective A Thesis presented to the Faculty department of Science & technology Studies of Cornell University (December 2003)

[17] Hansen, K.F., Bresin, R.: Mapping strategies in dj scratching. In: NIME '06: Proceedings of the 6th international conference on New interfaces for musical expression, Paris, France (2006) 188–191

[18] Falkenberg Hansen, K.: Playing the turntable: An introduction to scratching. Quarterly Progress and Status Report TMH-QPSR, Dept. for Speech and Hearing, KTH Computer Science and Communication (2001)

[19] Andersen, T.H.: In the mixxx: novel digital dj interfaces. In: CHI '05: CHI '05 extended abstracts on Human factors in computing systems, New York, NY, USA, ACM (2005) 1136–1137

[20] Jeffs, R.: Evolution of the dj mixer crossfader. A condensed version ran in DJ Times, June 1999 (1999) RaneNote 146 by Rane Corporation.

[21] Gwertzman, M.: With a nod to vynil, cd's take over the turntable. NY Times magazine (September 2004)

[22] Wong, W.: Storage and user interface for digital turntable. UQ Innovation Expo 2003 (October 2003)

[23] Williams, G.: The digital turntable. UQ Innovation Expo 2003 (October 2003)

[24] Hansen, K.F.: The basics of scratching. Journal of New Music Research **31**(4) (2002) 357–365

[25] Farrugia, R., Swiss, T.: Tracking the djs: Vinyl records, work, and the debate over new technologies. Journal of Popular Music Studies **17**(1) (2005) 30–44 Western Michigan University; University of Iowa.

[26] Gates, C., Subramanian, S., Gutwin, C.: Djs' perspectives on interaction and awareness in nightclubs. In: DIS '06: Proceedings of the 6th conference on Designing Interactive systems, New York, NY, USA, ACM (2006) 70–79

[27] Brewster, B., Broughton, F.: How to DJ Right: the art and science of playing records. Grove Press, New York, (2003)

[28] Nake, F.: Creativity in algorithmic art. In: C&C '09: Proceeding of the seventh ACM conference on Creativity and cognition, New York, NY, USA, ACM (2009) 97–106

[29] Levin, G., Lieberman, Z.: Sounds from shapes: audiovisual performance with hand silhouette contours in the manual input sessions. In: NIME '05: Proceedings of the 2005 conference on New interfaces for musical expression, Singapore, Singapore, National University of Singapore (2005) 115–120

[30] Hiller, L..L.I.: Experimental Music. Composition with an Electronic Computer. New York: McGraw-Hill. [Reprinted: Westport, CT: Greenwood Press, 1979.] (1959)

[31] Travis Pope, S.: Computer music workstations i have known and loved. In: Proc. of the International Computer Music Conference. (1995) 7

[32] Paradiso, J.A.: Electronic music: new ways to play. IEEE Spectr. **34**(12) (1997) 18–30

[33] Werde, B.: The d.j.'s new mix: Digital files and a turntable (October 2001) Digital Publication of NY Times Magazine, 25 October of 2001.

[34] Pennycook, B.W.: Computer-music interfaces: a survey. ACM Comput. Surv. **17**(2) (1985) 267–289

[35] Ableton Live 8 Official Documentation: `http://www.ableton.com/live-8` accessed on 20/11/2009.

[36] Cronin, D.: Feature into the groove: lessons from the desktop music revolution. interactions **15**(3) (2008) 72–78

[37] Pedersen, E.W., Hornbaek, K.: mixitui: a tangible sequencer for electronic live performances. In: TEI '09: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, New York, NY, USA, ACM (2009) 223–230

[38] Blackwell, A., Collins, N.: The programming language as a musical instrument. In: In Proceedings of PPIG05 (Psychology of Programming Interest Group. (2005) 120–130

[39] Propellerheads Rewire Documentation: `http://www.propellerheads.se/products/reason/index.cfm?fuseaction=get_article&article=rewire` accessed on 13/11/2009.

[40] Zadel, M., Scavone, G.: Laptop performance: Techniques, tools, and a new interface design. In: Proceedings of the International Computer Music Conference, New Orleans, LA, USA. Volume vol. 2006., Ann Arbor, MI: Scholarly Publishing Office, University of Michigan Library 2006 (2006)

[41] Zadel, M., Scavone, G.: Different strokes: a prototype software system for laptop performance and improvisation. In: NIME '06: Proceedings of the 2006 conference on New interfaces for musical expression, Paris, France, France, IRCAM — Centre Pompidou (2006) 168–171

[42] Loy, G.: Musicians make a standard: the midi phenomenon. Comput. Music J. **9**(4) (1985) 8–26

[43] Phillips, D.: State of the art: Linux audio 2008, part ii. Linux J. **2008**(174) (2008) 5

[44] Butler, M.J.: Unlocking the Groove: Rhythm, Meter, and Musical Design in Electronic Dance Music (Profiles in Popular Music). Indiana University Press (February 2006)

[45] Hellman, Z.F.: A look into the philosophy and technology that enable the disc jockey. Master of music, New York University (New York, NY) (2008) supervisor: Dr. Robert Rowe.

[46] Yerkes, K.M.: Survey of turntable-like hci for scrubbed media performance. Online at University of California, Santa Barbara (2009)

[47] Andersen, T.H.: Mixxx: towards novel dj interfaces. In: NIME '03: Proceedings of the 2003 conference on New interfaces for musical expression, Singapore, Singapore, National University of Singapore (2003) 30–35

[48] Faulkner, P., Runde, J.: Getting to grips with technology. Latter submited on a shorter version to Critical Realism: Progress and Challenges Conference (2004) Cambridge, Judge Business School and Girton College.

[49] Bowers, T.: Redefining spaces through its re-use: How to become the user of a space and not merely the function of it. Online at University of Plymouth School of Computing, Communications & Electronics (April 2006)

[50] Van Veen, T.: Digital djing: Richie hawtin and (the) final scratch. Discorder Jancember Magazine (2001)

[51] Peter Kirn: Ni-stanton final scratch divorce turns ugly; ni responds. Create Digital Music Magazine (February 2007)

[52] MsPinky Official Website and Documentation: `http://www.mspinky.com` accessed on 17/11/2009.

[53] Mixxx official Documentation: `http://www.mixxx.org` accessed on 10/11/2009.

[54] Mixxx Official Development Blog: `http://mixxxblog.blogspot.com/` accessed on 21/11/2009.

[55] Fitzmaurice, G.W., Ishii, H., Buxton, W.A.S.: Bricks: laying the foundations for graspable user interfaces. In: CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co. (1995) 442–449

[56] Patten, J., Recht, B., Ishii, H.: Audiopad: a tag-based interface for musical performance. In: NIME '02: Proceedings of the 2002 conference on New interfaces for musical expression, Singapore, Singapore, National University of Singapore (2002) 1–6

[57] Paradiso, J., Hsiao, K., Benbasat, A.: Tangible music interfaces using passive magnetic tags (2001)

[58] Paradiso, J.A., Pardue, L.S., Hsiao, K.Y., Benbasat, A.Y.: Electromagnetic tagging for electronic music interfaces. Journal of New Music Research **32.4** (December 2003) 28

[59] Patten, J., Recht, B., Ishii, H.: Interaction techniques for musical performance with tabletop tangible interfaces. In: ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology, New York, NY, USA, ACM (2006) 27

[60] Newton-Dunn, H., Nakano, H., Gibson, J.: Block jam. In: SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications, New York, NY, USA, ACM (2002) 67–67

[61] Newton-Dunn, H., Nakano, H., Gibson, J.: Block jam: a tangible interface for interactive music. In: NIME '03: Proceedings of the 2003 conference on New interfaces for musical expression, Singapore, Singapore, National University of Singapore (2003) 170–177

[62] Jordà, S.: Sonigraphical instruments from fmol to the reactable*. In: NIME '03: Proceedings of the 2003 conference on New interfaces for musical expression. (2003)

[63] Jordà, S., Geiger, G., Alonso, M., Kaltenbrunner, M.: The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In: TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction, New York, NY, USA, ACM (2007) 139–146

[64] : Max/msp version 5.0 music programming environment. Computer Music Journal **32**(2) (2008) 93 – 94

[65] Kaltenbrunner, M., Geiger, G., Jordà, S.: Dynamic patches for live musical performance. In: NIME '04: Proceedings of the 2004 conference on New interfaces for musical expression, Singapore, Singapore, National University of Singapore (2004) 19–22

[66] Hansen, K., Alonso, M., Dimitrov, S.: Combining dj scratching, tangible interfaces and a physics-based model of friction sounds. (2007)

[67] Hansen, K., Alonso, M.: More dj techniques on the reactable. In: NIME '08: Proceedings of the 2008 conference on New interfaces for musical expression. (2008)

[68] Hansen, K. - Skipproof Package Documentation and Download: `http://www.csc.kth.se/~kjetil/software.html` accessed on 02/10/2009.

[69] Villar, N., Lindsay, A.T., Gellersen, H.: Pin & play & perform: a rearrangeable interface for musical composition and performance. In: NIME '05: Proceedings of the 2005 conference on New interfaces for musical expression, Singapore, Singapore, National University of Singapore (2005) 188–191

[70] Schmidt, D., Block, F., Gellersen, H.: A comparison of direct and indirect multi-touch input for large surfaces. In: INTERACT '09: Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction, Berlin, Heidelberg, Springer-Verlag (2009) 582–594

[71] Mandoux, F., Wohlthat, S.: Design of a hci for skipproof". Technical report, 2F12213 Musical Communication and Music Technology (2004)

[72] Hans, M., Slayden, A., Smith, M., Banerjee, B., Gupta, A.: Djammer: a new digital, mobile, virtual, personal musical instrument. In: Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on. (July 2005) 4 pp.–

[73] Hayafuchi, K., Suzuki, K.: Musicglove: A wearable musical controller for massive media library. In: NIME '08: Proceedings of the 2008 conference on New interfaces for musical expression. (2008)

[74] Dipietro, L., Sabatini, A., Dario, P.: A survey of glove-based systems and their applications. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on

38(4) (jul. 2008) 461 –482

[75] Beamish, T., Doel, K.V.D., Maclean, K., Fels, S.: D'groove: A haptic turntable for digital audio control. In: in Proc. of ICAD. (2003)

[76] Villar, N., Gellersen, H., Jervis, M., Lang, A.: The colordex dj system: a new interface for live music mixing. In: NIME '07: Proceedings of the 7th international conference on New interfaces for musical expression, New York, NY, USA, ACM (2007) 264–269

[77] JazzMutant Official Website: `http://www.jazzmutant.com` accessed on 27/11/2009.

[78] JazzMutant Official Website, Lemur Objects: `http://www.jazzmutant.com/lemur_features_objects.php` accessed on 27/11/2009.

[79] Kobayashi, S., Akamatsu, M.: Spinner: a simple approach to reconfigurable user interfaces. In: NIME '05: Proceedings of the 2005 conference on New interfaces for musical expression, Singapore, Singapore, National University of Singapore (2005) 208–211

[80] Crevoisier, A.: Future-instruments.net: Towards the creation of hibrid electronic-acoustic musical instruments. In: Sonic Interaction Design: a Conf. Human Factors in Computing Systems (CHI) Workshop, Florence, Italy (April 2008) 53–58

[81] Kellum, G., Crevoisier, A.: A flexible mapping editor for multi-touch musical instruments. In: NIME '09: Proceedings of the 2009 conference on New interfaces for musical expression. (2009)

[82] Furnell, S., Dowland, P.: Advances in Networks, Computing and Communications 4. University of Plymouth (September 2007) ISBN: 978-1-84102-180-5.

[83] Davidson, P.L., Han, J.Y.: Synthesis and control on large scale multi-touch sensing displays. In: NIME '06: Proceedings of the 2006 conference on New interfaces for musical expression, Paris, France, France, IRCAM — Centre Pompidou (2006) 216–219

[84] : Products of interest: Lemur review. Computer Music Journal **29**(4) (2005) 99–112

[85] : Products of interest: The jazzmutant dexter daw interface. Computer Music Journal **31**(4) (2007) 88 – 101

[86] Izadi, S., Hodges, S., Butler, A., West, D., Rrustemi, A., Molloy, M., Buxton, W.: Thinsight: a thin form-factor interactive surface technology. Commun. ACM **52**(12) (2009) 90–98

[87] : Jazzmutant lemur review. Published in Sound on Sound Magazine (March 2007)

[88] Wright, M., Freed, A., Momeni, A.: Open sound control: State of the art 2003. In: International Conference on New Interfaces for Musical Expression, Montreal (2003) 153–159 OpenSound Control.

[89] JazzMutant Official Website, Lemur FAQ : `http://www.jazzmutant.com/faq_lemur_tech.php` accessed on 27/11/2009.

[90] Attigo Official Reports: `www.scotthobbs.co.uk` accessed on 30/11/2009.

[91] Fukuchi, K.: Multi-track scratch player on a multi-touch sensing device. In: ICEC 2007. Volume 4740. Springer (2007) 211–218

[92] Kin, K., Agrawala, M., DeRose, T.: Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In: GI '09: Proceedings of Graphics Interface 2009, Toronto, Ont., Canada, Canada, Canadian Information Processing Society (2009) 119–124

[93] Brandt, E., , Br, E., Dannenberg, R.: Low-latency music software using off-the-shelf operating systems. (1998)

[94] Levitin, D.J., Mathews, M.V., MacLean, K.: The perception of cross-modal simultaneity. In: Proceedings of International Journal of Computing Anticipatory Systems, Belgium. (1999)

[95] Wright, M., Dannenberg, R., Pope, S., Rodet, X., Serra, X., Wessel, D.: Panel: Standards from the computer music community. In: International Computer Music Conference, Miami, FL, International Computer Music Association (2004) 711–714

[96] Chaudhary, A.: Automated testing of open-source music software with open sound world and open sound control. In: ICMC 2005. (1/11/2005 2005)

[97] Wessel, D., Wright, M.: Problems and prospects for intimate musical control of computers. In: NIME '01: Proceedings of the 2001 conference on New interfaces for musical expression, Singapore, Singapore, National University of Singapore (2001) 1–4

[98] Moore, F.R.: The dysfunctions of midi. Comput. Music J. **12**(1) (1988) 19–28

[99] Bryan-Kinns, N.: Computers in support of musical expression. (Submitted to) ACM Computing Surveys (2004)

[100] Puckette, M.: Pure data: another integrated computer music environment. In: in Proceedings, International Computer Music Conference. (1996) 37–41

[101] Orlarey, Y., Fober, D., Letz, S.: Parallelization of audio applications with faust. In: Proceedings of the 6th Sound and Music Computing Conference, Porto, Portugal. (2009)

[102] Wright, M., Cassidy, R.J., Zbyszynski, M.F.: Audio and gesture latency measurements on linux and osx. In: International Computer Music Conference, Miami, FL, International Computer Music Association (2004) 423–429

[103] MacMillan, K., Droettboom, M., Fujinaga, I.: Audio latency measurements of desktop operating systems. International Computer Music Conference (2001)

[104] S.Letz, N.Arnaudov, R.Moret: What's new in jack2? In LAC, ed.: Proceedings of Linux Audio Conference 2009. (2009)

[105] Nacenta, M., Baudisch, P., Benko, H., Wilson, A.: Separability of spatial manipulations in multitouch interfaces. In: Graphics Interface, Kelowna, B.C., Canada (2009) 175–182

[106] Lopes, P., Fernandes, G., Jorge, J.A.: Trainable dtw-based classifier for recognizing feet-gestures. In: RECPAD 2010: Proceedings of the 16th Portuguese Conference on Pattern Recognition. (2010)

[107] Balakrishnan, R., Hinckley, K.: Symmetric bimanual interaction. In: CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2000) 33–40

A Sharing Mt-Djing files

To ease configurations, two XML formats were created to support the Mt-DJing application. The first is the most important towards DJs because it allows them to exchange different patches (the interface state with their objects), reinforcing cooperation and discussion upon the application. The latter eases configuration of the distributed modules, allowing the user to specify pairs of hosts and ports in which each module will be executed.

A.1 Sharing the Interface State

In order to share the interface state (or patch) the DJ just needs to load or save a new file. This file contains information about all objects used and connection between them. Inspiration is drawn directly from Pure Data internal text format, which works the same way, except it is not XML, thus is very difficult to edit it in text mode. A snippet of our format is shown below in A.1, each object is separated in its own XML node, and a scene file can hold as many scenes as a DJ wants. The application uses reflection to create all the objects dynamically from XML file.

Listing A.1: Snippet of Scene file

```
<Scene name="VimmiShow" author="PedroLopes">
  <objects>
    <SoundSource id="1" x="250" y="425" record="1"/>
    <Fader width="200" height="30" rotation="90" id="3" x="400" y="400"
       channel="1" start="1.0"/>
    <TripleEQ diameter="75" id="3" x="275" y="250" channel="1"/>
    <DrumPad x="525" y="225" scaleX="0.5" scaleY="0.5" id="0" numberPads="16"
      separatorNumber="4" padWidth="50" padHeight="50" padOffset="5"/>
    <MovableRecordsMenu x="0" y="0" id="0" width="200" height="2048"/>
  </objects>
 (...)
</Scene>
```

A.2 Sharing the Configurations

The XML file shown in A.2 details the configuration status of the various modules from which Mt-Djing is comprised. This allows to change ports and hosts, and module behaviour without any need for re-compiling. It also allows users to change modules, e.g.: running without Audio Module but having all the control data being sent via MIDI to an application, such as Ableton Live.

Listing A.2: Snippet of module configuration.

```
<module name="interface">
  <!--Interface is the visualization component where MtDjing will be rendered-->
    <input>
      <!--Interface is listening for messages of CoreApp in this host+port-->
      <host>localhost</host>
          <port>3000</port>
    </input>
    <output>
          <!--CoreApp is listening for messages of Interface in this host+port-->
          <host>localhost</host>
          <port>31339</port>
        </output>
    <options>
      <LoadScene name="VimmiShow">true</LoadScene>
    </options>
</module>
```

B DJ Appendixes

B.1 Selected Discography

The following music examples are landmarks in turntablist/scratch history, some highlighted by our accompanying group of DJs while others selected by the authors or drawn from DJ literature.

Kid Koala "Some of My Best Friends Are DJs" [2003]
C. Marclay, O. Yoshihide "Split 7"" [1999]
Birdy Nam Nam "Birdy Nam Nam" [2005]
M. Patton, X-Ecutioners "General Patton vs. The X-Ecutioners" [2005]
H. Hancock "Future Shock" [1983]
J. Mills "Blue Potential" [2006]

B.2 DJ Taxonomy

In Figure B.1 we present a possible taxonomy, drawn from an UML case model, that would clarify the reader about some DJ-styles. This is coherent with facts presented by Bell [5], Hansen [3] and Beamish's online taxonomy[1], although more detailed than the previous. This is not a final proposal, since it lacks various technical requirements and sub-styles such as Selectas. However, due to the void observed in the classification of DJs, this can suit the reader to a more in-depth understanding of our work.

From the taxonomy we can conclude that Mobile and Club-DJ cannot be viewed as a separate category in terms of actions/skills. The true differentiation parameter would be the type of setup used, and that is not featured in Figure B.1. Also, we observe that the Radio DJ has less technique requirements than the remaining but can be as high as Club-DJs, while the Scratch-DJ has the most advanced lexicon of actions/skills.

[1] `http://www.timothywisdom.com/science/djtaxonomy/djtaxonomy.php`, accessed on 12/12/2009.

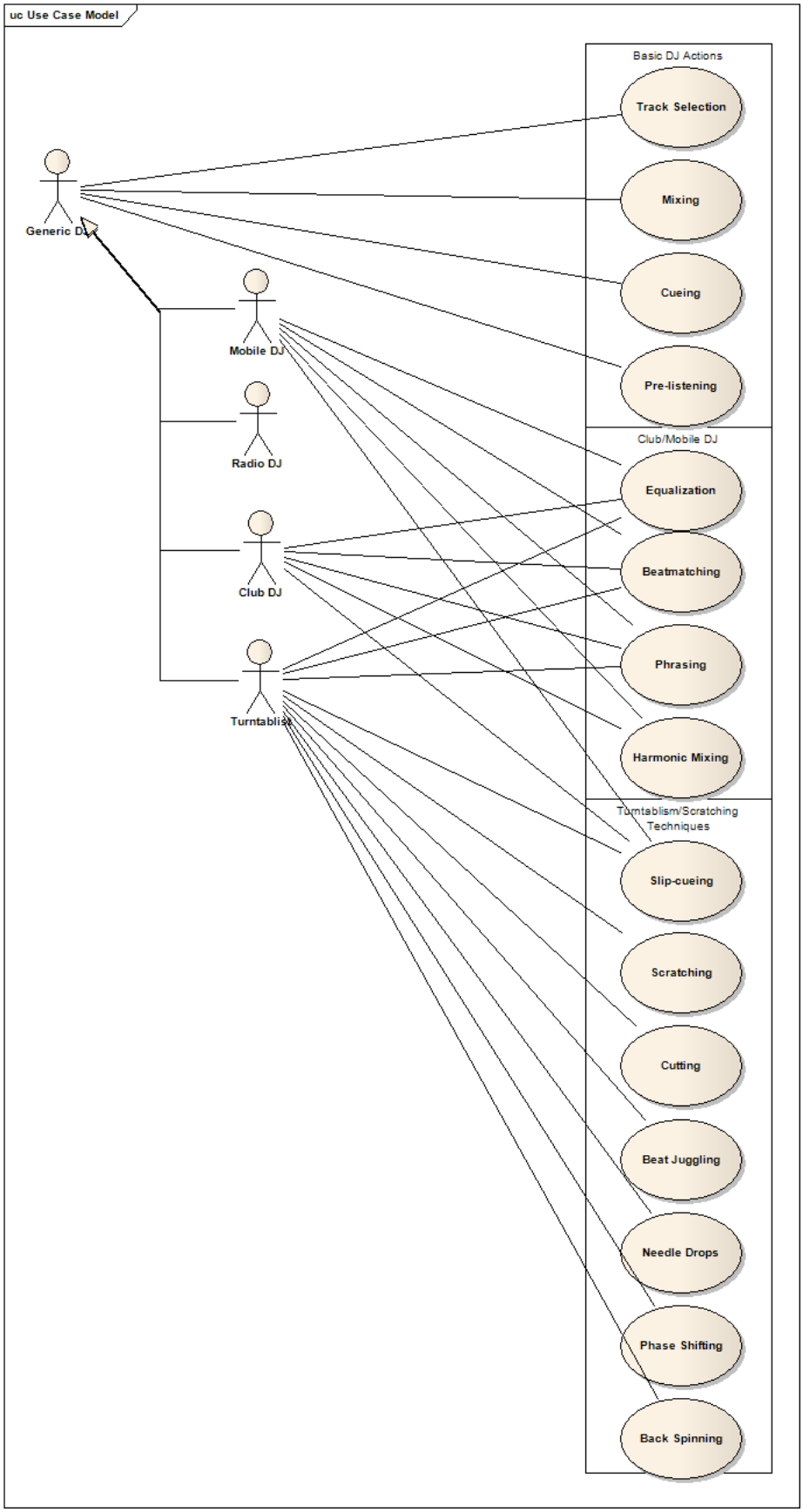


Figure B.1: Draft of DJ taxonomy.

B.3 Test setup specifications

B.3.1 Traditional Setup

We followed the typical setup found in DJ literature: two turntables and a mixer. For the turntables the best representative is the Technics models, from which we opted for two SL1200-mk5, which stand as "de facto" DJing standards. The mixer is an Urei 1600s, a full-fledged DJ mixer with all traditional features, suitable for comparison.

B.3.2 Virtual Setup

Mixxx was chosen as the virtual setup, due to three factors. Firstly because it is an open source software; secondly because it represents a typical Djing application based on the dual turntable metaphor and lastly none of our users had any experience with it. Thus it posed as a suitable option for validating learning curves. The focus of our work was not criticizing Mixxx but rather having a strong comparison metric between a Virtual setup and our solution.

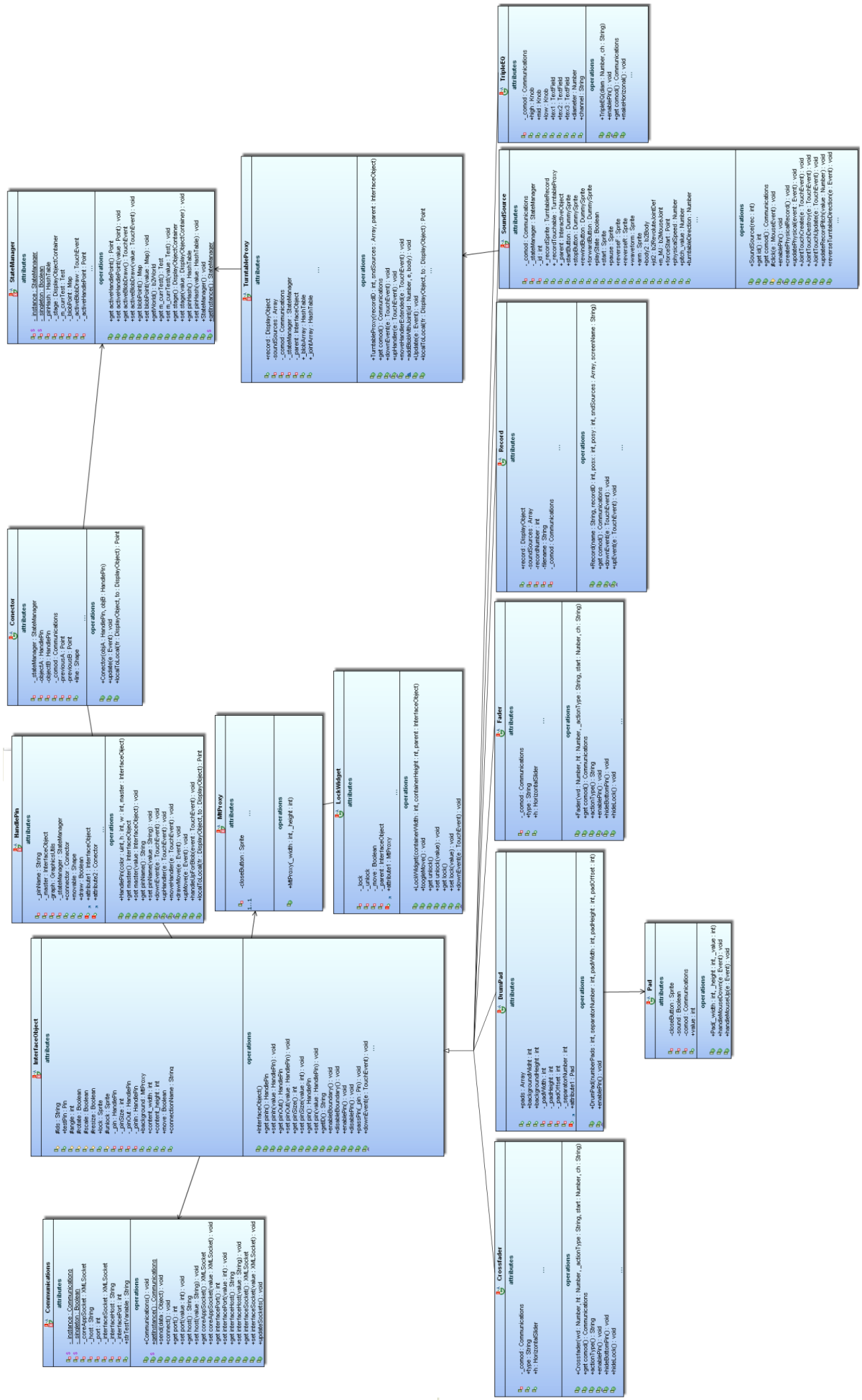
B.3.3 Hybrid Setup

Again, Mixxx served this purpose. It is capable of receiving tracking information and posing as a hybrid setup when coupled to our aforementioned Traditional. We used an off-the-shelf USB soundcard to deliver the sound back again to the mixer, ensuring low latency. Furthermore, likewise as for the Virtual, we wanted the DJs to use a Hybrid solution that they were not acquainted to.

C Implementation Details

C.1 UML Class Diagram

The represented class diagram shows detailed view over the general architecture of objects. in order to keep the diagram readable, several objects, debug properties and operators were hidden from the classes.


StateManager
attributes
-instance : StateManager
-stateMap : HashTable
-pinHash : HashTable
-_stage : DisplayObjectContainer
-m_curr Test : Test
-_lastBlockPin : Pin
-_activeBlockDraw : TouchEvent
-_activeHandePoint : Point
operations
+get activeHandePoint() : Point
+set activeHandePoint(value : Point) : void
+get activeBlockDraw() : TouchEvent
+set activeBlockDraw(value : TouchEvent) : void
+get blockPoint() : Map
+set blockPoint(value : Map) : void
+getWorld() : b2World
+setWorld(value : b2World) : void
+set m_curr test(value : Test) : void
+get stage() : DisplayObjectContainer
+setStage(value : DisplayObjectContainer) : void
+get pinHash() : HashTable
+set pinHash(value : HashTable) : void
+StateManager() : void
+getInstance() : StateManager
Conector
attributes
-_stateManager : StateManager
-objectA : HandlePin
-objectB : HandlePin
-_comod : Communications
-previousA : Point
-previousB : Point
-line : Shape
...
operations
+Conector(obA : HandlePin, obB : HandlePin)
+updateLine(e : Event) : void
+localToLocal(fr : DisplayObject, to : DisplayObject) : Point
TurntableProxy
attributes
+record : DisplayObject
-soundSources : Array
-_comod : Communications
-_stateManager : StateManager
-_parent : InteractiveObject
-_jBlockArray : HashTable
-_jLinkArray : HashTable
...
operations
+TurntableProxy(record : int, sndSources : Array, parent : InteractiveObject)
+get comod() : Communications
+downEvent(e : TouchEvent) : void
+upHandler(e : TouchEvent) : void
+upHandleRecordEvent(e : TouchEvent) : void
+addBlockWhenJoint(id : Number, e, body) : void
+Update(e : Event) : void
+localToLocal(fr : DisplayObject, to : DisplayObject) : Point
...
TripleEQ
attributes
-_comod : Communications
-high : Knob
-mid : Knob
-low : Knob
-text1 : TextField
-text2 : TextField
-text3 : TextField
-channel : Number
-channel : String
operations
+TripleEQ(dam : Number, ch : String)
+enablePri() : void
+get comod() : Communications
+initializeScreen() : void
SoundSource
attributes
-_comod : Communications
-_stateManager : StateManager
-_id : int
+_recordSprite : TurntableRecord
+_parent : InteractiveProxy
+_parent : InteractiveObject
+startButton : DumnySprite
+stopbutton : DumnySprite
+_pitchButton : DumnySprite
+forwardButton : DumnySprite
+playState : Boolean
+_pitch : Sprite
+pause : Sprite
+reverseF : Sprite
+reverseR : Sprite
+_pitchSlider : Sprite
+arm : Sprite
+body2 : b2Body
+_mouseJoint : b2MouseJoint
+m_MJ : b2MouseJoint
+forceSet : Point
+_pitchValue : Number
+pitch : value : Number
+turntableDirection : Number
...
operations
+SoundSource(rec : int)
+get id() : int
+get comod() : Communications
+initializeScreen() : void
+enablePri() : void
+createPhysicalRecord() : void
+updateRecordEvent(e : Event) : void
+downTouchDestroy(e : TouchEvent) : void
+upTouchDestroy(e : TouchEvent) : void
+updateRecordPitch(value : Number) : void
+reverseTurntableDirection(e : Event) : void
...
Record
attributes
+record : DisplayObject
-soundSources : Array
-recordNumber : int
-_screenName : String
-_comod : Communications
...
operations
+Record(name : String, recordID : int, posx : int, posy : int, sndSources : Array, screenName : String)
+get comod() : Communications
+downEvent(e : TouchEvent) : void
+upEvent(e : TouchEvent) : void
...
Fader
attributes
-_comod : Communications
+type : String
+h : HorizontalSlider
...
operations
+Fader(vid : Number, ht : Number, _actionType : String, start : Number, ch : String)
+get comod() : Communications
+initialize() : void
+enablePri() : void
+hideButtonPri() : void
+hideLock() : void
...
HandlePin
attributes
-_pinName : String
-_master : InteractiveObject
-graph : GraphicsUtils
-_stateManager : StateManager
+connector : Conector
+movable : Shape
+_pinShape : Shape
+ attribute1 : InteractiveObject
+ attribute2 : Conector
...
operations
+HandlePin(color : uint, x : int, y : int, master : InteractiveObject)
+get master() : InteractiveObject
+set master(value : InteractiveObject) : void
+get pinName() : String
+set pinName(value : String) : void
+downEvent(e : TouchEvent) : void
+upHandler(e : TouchEvent) : void
+upHandleMoveEvent(e : TouchEvent) : void
+drawMove(e : Event) : void
+upMove(e : Event) : void
+downHandleMoveEvent(e : TouchEvent) : void
+localToLocal(fr : DisplayObject, to : DisplayObject) : Point
MPrxy
attributes
-closeButton : Sprite
...
operations
+MPrxy(_width : int, _height : int)
LockWidget
attributes
-_lock
-_unlock
-_move : Boolean
-_parent : InteractiveObject
+ attribute1 : MPrxy
...
operations
+LockWidget(containerWidth : int, containerHeight : int, parent : InteractiveObject)
+closeWidget() : void
+get lock()
+set unlock(value) : void
+get lock()
+set lock(value) : void
+downEvent(e : TouchEvent) : void
InterfaceObject
attributes
#title : String
+leadPin : Pin
#height : int
#visible : Boolean
#scale : Boolean
#rotate : Boolean
#_text : TextField
#unlock : Sprite
-_pin : HandlePin
-_pinOut : HandlePin
-_pinIn : HandlePin
+backgroundMPrxy
+contentWidth : int
+contentHeight : int
+move : Boolean
+connectorNetwork : String
...
operations
+InterfaceObject()
+get pinIn() : HandlePin
+set pinIn(value : HandlePin) : void
+get pinOut() : HandlePin
+set pinOut(value : HandlePin) : void
+get pinSize() : int
+set pinSize(value : int) : void
+get pin() : HandlePin
+set pin(value : HandlePin) : void
+getLock() : Sprite
+setLock(value) : void
+disableBoundary() : void
+enablePri() : void
+hideBtnPri() : void
+passPin(_pin : Pin) : void
+downEvent(e : TouchEvent) : void
...
Communications
attributes
-instance : Communications
-isinstance : Boolean
-_conSocket : XMLSocket
-_port : String
-_host : String
-_port : int
-_interfaceSocket : XMLSocket
-_interfaceHost : String
-_interfacePort : int
+frTestVariable : String
...
operations
+Communications() : void
+getInstance() : Communications
+getSocket() : XMLSocket
+connect() : void
+get port() : int
+set port(value : int) : void
+get host() : String
+set host(value : String) : void
+get coreAppSocket() : XMLSocket
+setCoreAppSocket(value : XMLSocket) : void
+get interfacePort() : int
+set interfacePort(value : int) : void
+get interfaceHost() : String
+set interfaceHost(value : String) : void
+get interfaceSocket() : XMLSocket
+set interfaceSocket(value : XMLSocket) : void
+getNetworkIP() : void
...
DrumPad
attributes
+pads : Array
+backgroundWidth : int
+backgroundHeight : int
-_padWidth : int
-_padHeight : int
-_padsOffset : int
-_separationNumber : int
+ attribute1 : Pad
...
operations
+DrumPad(numberPads : int, separationNumber : int, padWidth : int, padHeight : int, padOffset : int)
+enablePri() : void
...
Pad
attributes
-closeButton : Sprite
-sound : Boolean
-_comod : Communications
+value : int
...
operations
+Pad(_width : int, _height : int, _value : int)
+handleMouseDown(e : Event) : void
+handleMouseUp(e : Event) : void
Crossfader
attributes
-_comod : Communications
+type : String
+h : HorizontalSlider
...
operations
+Crossfader(vid : Number, ht : Number, _actionType : String, start : Number, ch : String)
+get comod() : Communications
+initialize() : void
+enablePri() : void
+hideButtonPri() : void
+hideLock() : void
...

D DJ Terminology

AD Analogue to digital signal conversion.

Automation Refers to the ability of drawing/recording a behaviour for a certain parameter in time. It can then be applied as a sequence in realtime.

Backspin Refers to spinning a record backwards for a short and quick spurt.

Beat estimation Mechanism for detecting the BPM of a song automatically.

Beat-juggling Fast mix of a sample, in order to extend its duration, by looping it manually with a copy of the same sample. Typically playing it from one turntable to the other.

Beatmatching Aligning two different tracks simultaneously and producing a new composition in real-time.

BPM Beats Per Minute, describes the musical *tempo* of a song.

Club DJ DJ that usually works with electronic/dance music for crowds.

Cross-fader Sliding potentiometer device on the mixing board that allows the DJ to transfer the audio output from one source to another.

Cueing Act of finding the correct starting point of a track.

Curve Control Defines the behaviour of the sliding faders along their path.

DA Digital to analogue signal conversion (converse operation to AD) .

DJ Disc-Jockey.

DJ Mixer Desk that allows DJs to combine multiple incoming audio signals.

DJing Activity of the DJ, mixing and manipulating music tracks in a live session.

Dropping a track Mixing a new track into the DJ performance.

EQ (or Equalizer) is a component that shapes the frequencies of a signal.

Equalization Denotes the task of applying EQ in realtime during a mix.

Fader-start Technology that allows the connection of a CD player to a crossfader, such that the DJ can use the crossfader to trigger the CD-player start button automatically.

Hybrid Setup Comprised of a Virtual setup that can be controlled with Traditional gear, such as real turntables or CD-players.

Hot-cue Terminology used in CD-players to denote a stored position of the track where a DJ can jump to, instantly.

Insert Denotes a a point where the audio signal path can be altered/routed.

Knob Potentiometers that alter parameters of the mixing desks and other gear.

Matrix routing Allows any input of a mixing desk to be mapped to any channel/s.

MIDI Musical Instrument Digital Interface, a protocol for controlling software/instruments.

Mixing Blending two audio sources together in one homogeneous new sound.

Mobile DJ Very similar to the Club DJ, except it usually plays in different locations on a regular basis.

OSC Open Sound Control, protocol for controlling messages.

Panning Effect of having a sound moving in the stereoscopic field.

Pre-listening Preview a track before it goes to the main output/audience.

Pitch Slider that allows adjustment in the speed of the turntable/CD motion, which causes the sound's musical pitch to increase or decrease.

Plugin Smaller application that can be hosted inside the main audio software.

Quantitize Subdividing a musical tempo in smaller ones. A sample is also called "quantitized" when it is placed on a correct timing of the musical tempo/BPM.

Radio DJ DJ that works in radio stations, blending music with talk shows.

RPM Revolutions (or rotations) per minute, denotes a speed in which a recording was pressed or is listened.

Sampler Digital recording device, which is usually able to loop the sampled sounds after recording is made.

Scratching Moving the record back and forth, causing the sound to be played forwards and backwards at various speeds. It is usually paired with crossfader combinations.

Send/Return (or Dry/Wet) Amount of signal that is routed to/from an effects processor.

Setup The gear that a DJ uses for his/her performance.

Slip-cueing Holding a record still while the platter rotates underneath and then releasing it at the right moment.

Stylus Or needle, is the pick-up that captures the sound recorded into a vinyl record.

Tempo-lock Alter a song's musical pitch without altering the tempo.

Timestretch (or Digital key correction or Keylock) Change a song tempo without changing its musical pitch.

Traditional Setup Traditional DJ gear, comprised of analogue turntables, CD-players and a mixer.

Trim/gain Control for decreasing/increasing an input volume on the mixing desk.

Turntablist DJ that does heavily scratch-based acts.

Virtual Setup Software application that allows DJing with digital audio files.

VU Volume Units, refers to a meter where the amplitude can be measured.